AFHRL-TP-90-10

2

AIR FORCE HUMAN RESOURCES LABORATORY

CONTENT DATA MODEL: TECHNICAL SUMMARY

Mark Earl, Captain, USAF
David R. Gunning

LOGISTICS AND HUMAN FACTORS DIVISION
Wright-Patterson Air Force Base, Ohio 45433-6503

Eric Freese
Ronald Shroder
Walter Werts
Tim Reser

RJO Enterprises, Inc.
101 Woodman Drive
Dayton, Ohio 45431

May 1990

Final Technical Paper for Period August 1988 - January 1990

Approved for public release; distribution is unlimited.

AIR FORCE SYSTEMS COMMAND
BROOKS AIR FORCE BASE, TEXAS 78235-5601

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility or any obligation whatsoever. The fact that the Government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise in any manner construed, as licensing the holder, or any other person or corporation; or as conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

The Public Affairs Office has reviewed this paper, and it is releasable to the National Technical Information Service, where it will be available to the general public, including foreign nationals.

This paper has been reviewed and is approved for publication.

BRYAN K. CAPORLETTE, 2Lt, USAF
Contract Monitor

BERTRAM W. CREAM, Technical Director
Logistics and Human Factors Division

JAMES C. CLARK, Colonel, USAF
Chief, Logistics and Human Factors Division

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
May 1990

3. REPORT TYPE AND DATES COVERED
Final August 1988 - January 1990

4. TITLE AND SUBTITLE
Content Data Model: Technical Summary

5. FUNDING NUMBERS
C - F09603-87-G-0659
PE - 63106F
PR - 2950
TA - 00
WU - 08

6. AUTHOR(S)
Mark Earl
Eric Freese
Walter Werts
David R. Gunning
Ronald Shroder
Tim Reser

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
RJO Enterprises, Incorporated
101 Woodman Drive
Dayton, Ohio 45431

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Logistics and Human Factors Division
Air Force Human Resources Laboratory
Wright-Patterson Air Force Base, Ohio 45433-6503

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFHRL-TP-90-10

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Due to the increasing complexity and number of modern weapon systems, the Air Force faces an ever-growing number of paper-based technical orders. The Air Force Human Resources Laboratory has conducted research and development (R&D) of automated technical information systems. This research investigated technologies for the storage, distribution, and presentation of technical information. Benefits of this research will include improvement in the performance of maintenance personnel and reduction in the cost of maintaining Air Force technical information.

One of the products of this R&D includes a technology which provides the Air Force with an economical way of storing and presenting technical data. The Content Data Model (CDM) is a specification for a data base which is intended to store all of the technical information for a weapon system. At this time, the CDM stores only maintenance and operational information. This paper presents: (a) a description of the work involved in the development of the CDM, (b) a detailed description of the CDM, (c) efforts to demonstrate and validate the CDM, and (d) the development of a draft specification for the preparation and delivery of technical data in CDM format.

14. SUBJECT TERMS
computer-aided acquisition and logistics support system (CALS) ,
content data model
standard generalized markup language
digital technical information

15. NUMBER OF PAGES
152

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT
Unclassified

20. LIMITATION OF ABSTRACT
UL

AFHRL Technical Paper 90-10 May 1990

CONTENT DATA MODEL: TECHNICAL SUMMARY

Mark Earl, Captain, USAF
David R. Gunning

LOGISTICS AND HUMAN FACTORS DIVISION
Wright-Patterson Air Force Base, Ohio 45433-6503

Eric Freese
Ronald Shroder
Walter Werts
Tim Reser

RJO Enterprises, Inc.
101 Woodman Drive
Dayton, Ohio 45431

Reviewed by

Robert C. Johnson
Chief, Cɔmbat Logistics Branch

Submitted for publication by

Bertrem W. Cream, Technical Director
. Logis ˋnd Human Factors Division

This publication is primarily a working paper. It is published soley to document work performed.

SUMMARY

Due to the increasing complexity and number of modern weapon systems, the Air Force faces an ever growing number of paper-based technical orders. The Air Force Human Resources Laboratory has conducted research and development of automated technical information systems. This research investigated technologies for the storage, distribution, and presentation of technical information. Benefits of this research will include improvement of performance of maintenance personnel and reduction in cost of maintaining Air Force technical information.

One of the products of this research includes a technology which provides the Air Force with an economical way of storing and presenting technical data. The Content Data Model (CDM) is a specification for a data base which is intended to store all of the technical information for a weapon system. At this time, the CDM stores only maintenance and operational information. The CDM will facilitate the interchange of technical information between contractors and the government.

This paper presents:

- A description of the work involved in the development of the CDM.
- A detailed description of the CDM.
- Efforts to demonstrate and validate the CDM.
- The development of a draft specification for the preparation and delivery of technical data in CDM format.

PREFACE

The work described in this paper was performed in support of the Computer-Aided Acquisition and Logistic Support initiative of the Air Force Systems Command. This work is consistent with the Air Force Human Resources Laboratory, (AFHRL) Logistics and Human Factors Division's Mission to improve the combat maintenance capability and readiness of the Air Force.

The work described in this paper was performed by the Air Force Human Resources Laboratory, RJO Enterprises, Inc., and Datalogics, Inc. The RJO effort was conducted under contract #F09603-87-G-0659/SG02; the Datalogic's work was performed under subcontract to RJO. Captain Mark J. Earl, AFHRL Combat Logistics Branch (LRC), served as the laboratory technical monitor and the contracting officer's technical representative (COTR) for the effort. Major contributors to this effort included Mr. David Gunning (AFHRL) and Major Paul Condit Air Force Logistics Command (AFLC).

TABLE OF CONTENTS

LIST OF FIGURES

THE CONTENT DATA MODEL: TECHNICAL SUMMARY

1. INTRODUCTION

Currently a number of separate technical manuals are required to support a system or subsystem. Each manual is designed to support specific maintenance functions and contains the information required to accomplish those actions. Examination of the manuals for typical system reveals that, in many instances, a given piece of information occurs in more than one manual and sometimes several times in the same manual. For example, a warning may apply to several maintenance actions and appear in several manuals or several times in the same manual (see Figure 1). This redundancy is inevitable in a paper based system. However, the use of a fully digital technical data system will provide a means of eliminating this unnecessary redundancy and greatly reduce storage requirements, simplify the update process, and insure more up to date technical data. The use of a fully integrated data base that contains all of the technical information required to maintain a system will make this possible. The integrated data base will store each piece of information once. Then, when a procedure is called which uses that information, the information is extracted from the data base and combined with other elements of information to construct the exact information (e.g., a step-by-step procedure) for presentation on a computer display or for printing on paper as needed. This paper describes a research effort conducted by the Air Force Human Resources Laboratory to develop and test a model of an integrated data base to accomplish this objective. The model developed is known as the Content Data Model (CDM). The development and validation of the CDM is described in the following sections.

Objective

The basic objective of this effort was to develop and validate a model of an integrated data base to maintain technical data to support maintenance operations. Specific objectives of the effort were to:

1. Design and develop the integrated data base model (the CDM).
2. Analyze technical data requirements and compare the results of the analysis with the CDM to insure that the CDM adequately supports all technical information requirements.
3. Compare the CDM with exiting technical information systems and standards to insure compatibility.
4. Develop a specification defining requirements for technical data developed for use with the CDM.
5. Validate the CDM specification by designing and building a data base, populating the data base with sample data, and extracting the data for presentation in different formats on different media.

II. BACKGROUND

In order to support the CALS initiative, several organizations within the DoD and industry conducted efforts to develop technical information development and delivery systems. These groups included AFHRL, the Air Force Logistics Command (AFLC), the

Navy, the Army, and, Northrop Corporation, the prime contractor for the B-2 bomber. This section highlights the efforts undertaken by AFHRL and provides a background to the development of the CDM. Also examined are three relevant forerunners of the CDM: Automated Technical Order System (ATOS), Improved Technical Data System (ITDS), and MIL-M-28001.

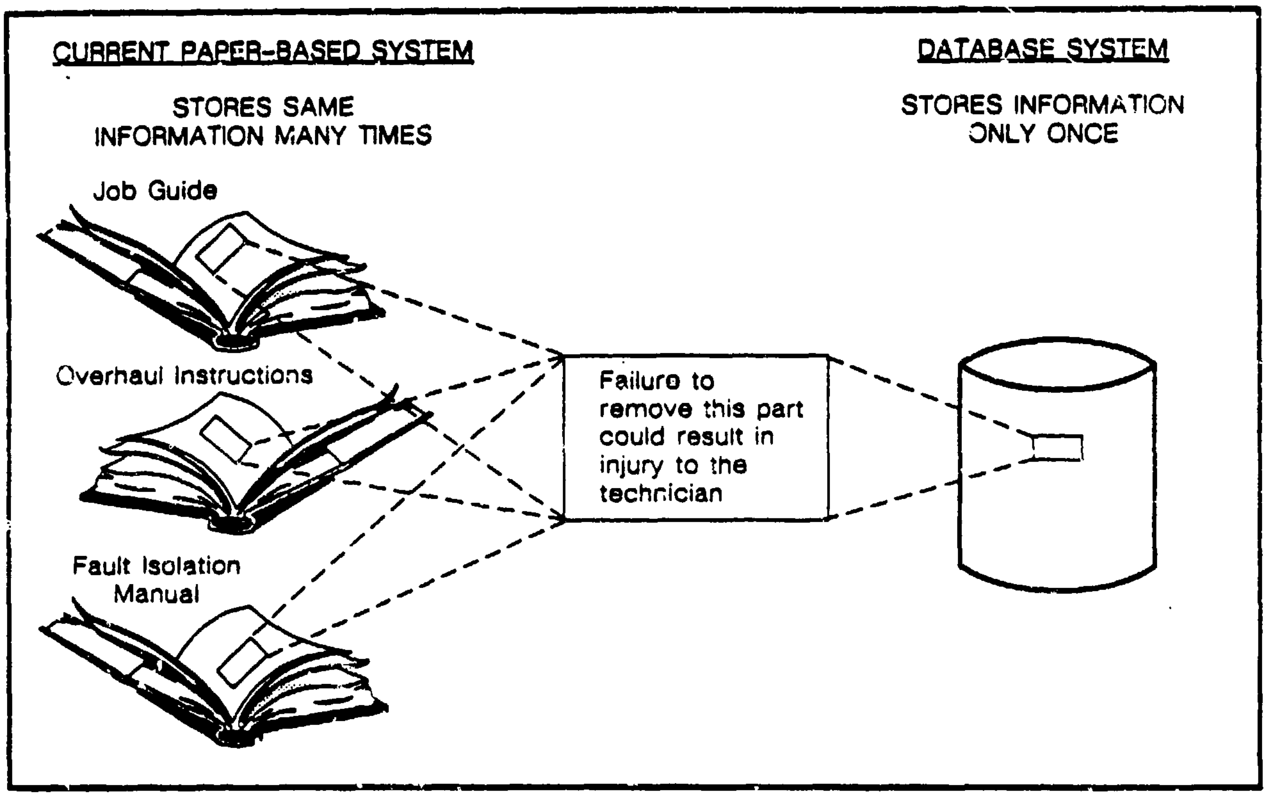

CURRENT PAPER-BASED SYSTEM
STORES SAME INFORMATION MANY TIMES
Job Guide
Overhaul Instructions
Fault Isolation Manual
Failure to remove this part could result in injury to the technician
DATABASE SYSTEM
STORES INFORMATION ONLY ONCE

Figure 1. **Reduction of Storage Requirements.**

AFHRL Activities

In 1976, AFHRL began conducting research and development (R&D) of automated technical information systems to support aircraft maintenance technicians. This R&D investigated technologies for the storage, distribution and presentation of technical information using automated systems. The research was done to improve the performance and productivity of maintenance personnel and to reduce the cost of maintaining Air Force technical information. The research has progressed through four increments (see Figure 2):

1. Computer-Based Maintenance Aiding System, which tested information presentation and human-computer interaction techniques.

2. Authoring and Presentation System, which refined the definition of data elements as well as providing a means for authoring and presenting technical information.

3. Integrated Data Base, which was designed to store all of the information for a particular weapon system.

4. **Content Data Model, which facilitates the interchange of technical information and provides a structure for the storage of technical information.**

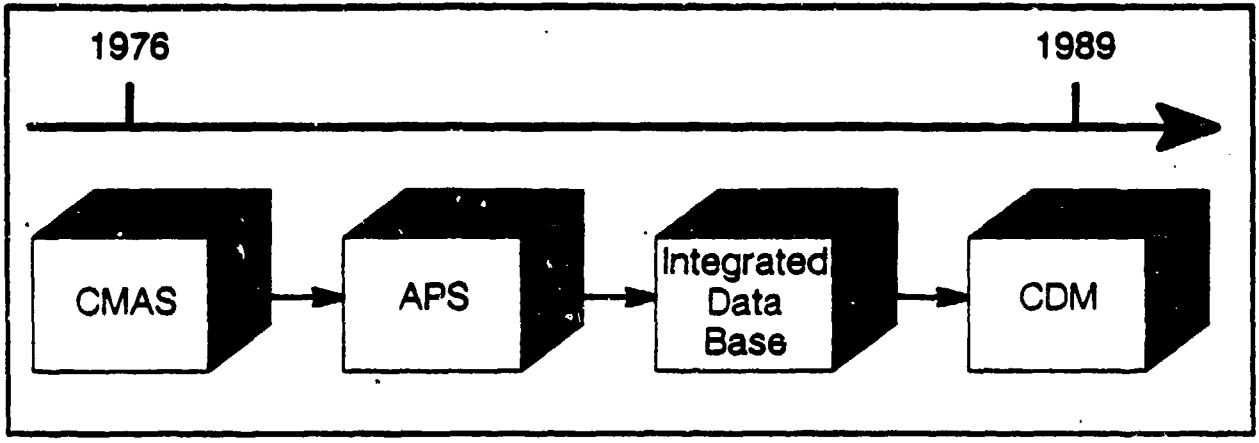


Figure 2. Related AFHRL Activities.

Computer-Based Maintenance Aiding System

AFHRL developed the Computer-Based Maintenance Aiding System (CMAS) to evaluate integrated information presentation and human/computer interface techniques for technical information. The CMAS systems also allowed investigation into the use of digital technical information under controlled field conditions. Specific concepts evaluated included multiple levels of detail, various facilities for accessing technical manual data, presentation of diagrams larger than the screen, function key utility, human/computer interaction, and presentation of troubleshooting procedures.

Two prototype systems for intermediate level maintenance were developed under the CMAS program. Among the many lessons learned from the project was that a more efficient means of developing the technical data to be presented on an automated system was required. The process of developing technical information for the prototypes required the use of complex codes to control the retrieval and presentation of the data. Technical data authors were required to input these complex codes by hand with only limited assistance from authoring aids. The result was that the creation of the data was expensive, and the data contained many system control code errors. Also, the technical data and technical data presentation software developed for CMAS was operational on only one computer system. This highlighted the necessity of designing a data base which would not be limited to use on one system. These observations, along with similar concerns expressed by AFHRL personnel working in support of the AFLC Automated Technical Order System (ATOS), led to a decision to investigate the storage of technical information in a data base.

A contract with Rockwell was modified to include a requirement to conduct an analysis of the data base requirements and to develop a coding scheme for preparing data for storage in a "neutral" data base. Neutral data is designed so that it can be presented on any computer. The preliminary study performed by Rockwell included a detailed analysis of the Air Force specifications for intermediate level technical data. The analysis of the data identified each element of data and the characteristics of each element. The characteristics (or attributes) were then categorized according to elements with common attributes. These categories provided the necessary source information for the development of a preliminary data base design. The design (or coding scheme) was developed under subcontract by Datalogics, Inc. It was based upon the Standard Generalized Markup Language (SGML) which was also being used in the ATOS program. SGML was selected as a starting point to ensure data base compatibility with ATOS. The SGML coding scheme labels each element of data (e.g., a paragraph, part number, or title) with an identifier specifying the type of information and relationships to other data elements. For example, a paragraph may have codes specifying the chapter it belongs to, the preceding paragraph, or an illustration which accompanies the paragraph.

Once the technical information, with its corresponding codes, is developed and stored, it can be formatted for printing or for display on a computer monitor. Special software formats the data according to the characteristics of the specific display device and predefined formatting rules. For example, if a data element is a header element, presentation (or printing) of the data will occur in the location on the display (or page) specified by the formatting rules and display (or page) characteristics. This approach makes it possible to display technical data on a variety of systems without modification to the data base. It also makes it possible to print the data, if desired, or use it in other ways without modifying the data base.

Authoring and Presentation System

The results of the analysis and coding specification provided the starting point for an AFHRL in-house effort to develop an efficient means of developing data in a neutral format for presentation on a variety of systems. The primary objectives of this new effort were to extend and refine the set of identified data elements and to develop software for a system to author and present technical data using a neutral data base. The Authoring and Presentation System (APS) effort developed a neutral data base which could be presented on various displays and printed according to specified formats without requiring changes in the data base (See Figure 3).

The Authoring and Presentation System effort developed software to provide an efficient means of creating the data with the requirement for the neutral data base and software to present the data on various computer systems.

The design of the APS data bases was initially based on paper technical data. Instead of duplicating the structure of the current technical order system, a data design and data base approach designed primarily for use by electronic applications was developed.

The authoring portion of the system allowed a technical manual author to create data as if the writer were using a standard word processing system. Using this part of the system, the author could

1. create or modify data.

2. attach attributes (e.g., this is a paragraph).

3. link supporting information to the data.

Document variations and configuration control were also maintained. This allowed the author to maintain many versions of a document from within the same data base. The presentation portion of APS displayed only the information needed by a particular maintenance technician for the particular aircraft and the specific task on which the technician was working.

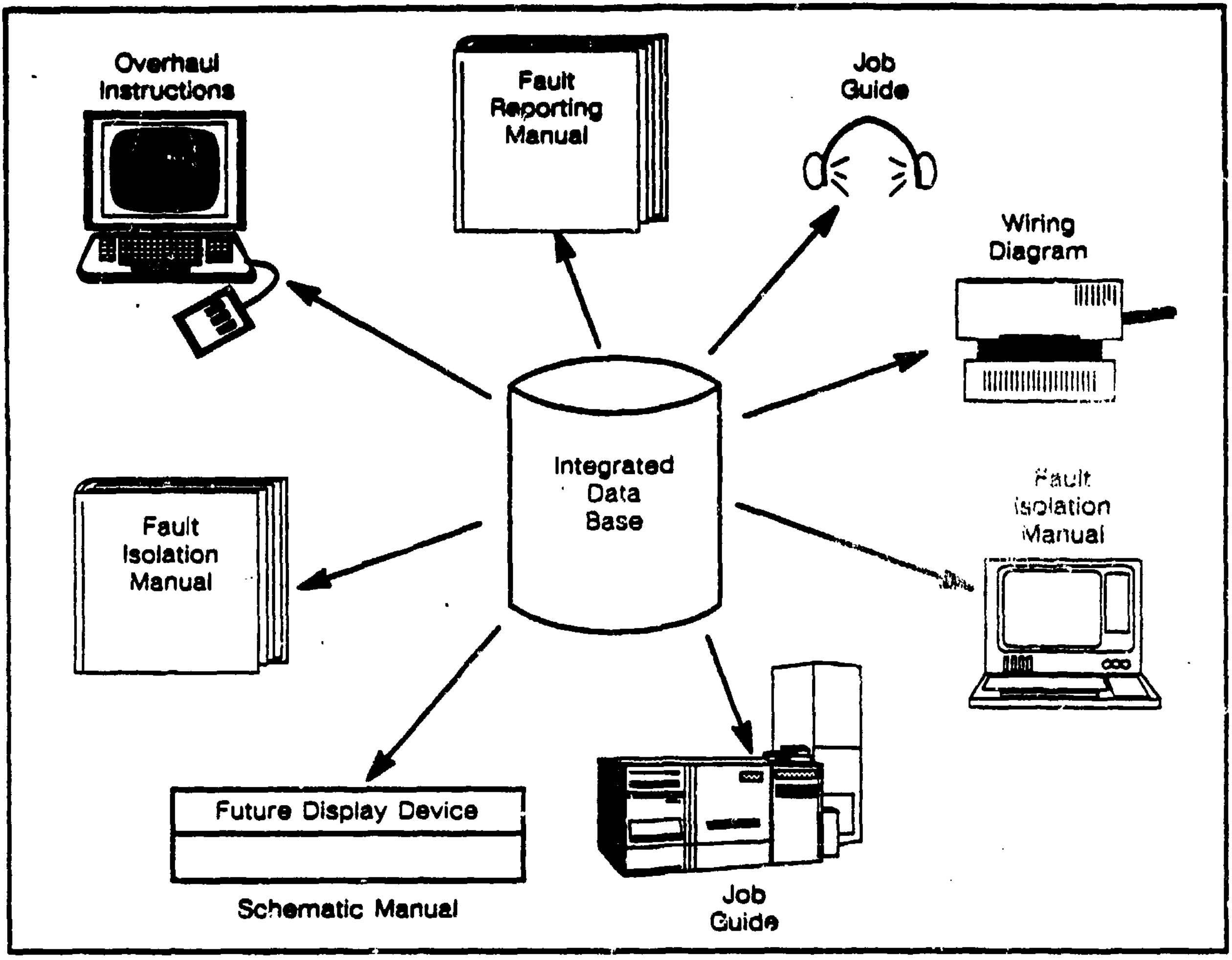


Figure 3. Present and Future Display Technologies.

APS provided a way to author and present data in a neutral format. However, each different Technical Order (TO) had its own data base resulting in redundancy of data since information used in several TOs was repeated in each Technical Order. The concept of an integrated data base was developed to model all of the technical information for a weapon system in a single neutral data base while removing the redundant data.

Integrated Data Base Design

The integrated technical information data base contains all of the information for a particular weapon system. While APS stored technical information manual by manual, the integrated data base stores many manuals in a single data base. Such a data base is considered integrated because different types of information (i.e., organizational, intermediate, depot) found in different manuals can be stored within the one data base. The information within the data base can be cross-referenced similar to the way in which cross-references are made in the present manuals. For example, if in the process of doing a fault isolation procedure a reference is made to a job guide procedure, the technician using the current paper TOs would have to find the correct job guide manual, complete the procedure, and then return to the fault isolation manual. The same technician accessing the data from the integrated data base would be able to access the information automatically. Once the job guide procedure has been completed, the technician would be automatically returned to the same place in the fault isolation procedure and continue working.

Each piece of technical information consists of three items: a) content, b) structure, and c) format (see Figure 4). Content is the actual information which is presented. The content of the information is interpreted by the reader as it is accessed. Structure consists of the organization of the technical information and sequence in which the technical information will be displayed. For example, a task in a technical manual consists of steps, illustrations, warnings, notes, cautions, etc. The technical information within a technical manual may be sequenced in a specific order such as an introduction followed by the steps with warnings, notes, and cautions where appropriate. Format is the appearance and style on the displayed information. For example, numbers may be added to the steps or a warning may be labelled with the word "warning" within a box which is centered on the page.

Data for presentation on automated technical data systems may be stored with or without formatting information. It is more efficient to store the data without formatting for two reasons: a) data with formatting information can only be presented on specific computers, and b) formatted data must be stored with its unique format requirements every time it is used creating redundancy. The use of an integrated data base with neutral, unformatted data overcomes both problems. The data is stored only once no matter how many times it may be used. For example, a warning may be stored only once even though it is used in many procedures.

The data bases of technical information will also be easier to maintain than the present technical order libraries. Required changes and updates can be quickly incorporated into the main data base since the data is stored once. This ensures that the users have access to the most accurate information available.

Content Data Model (CDM)

The requirement for a neutral, integrated data base resulted in the development of the CDM. The CDM is a model for the interchange of integrated data bases of technical information.

Data developed using the CDM structure will

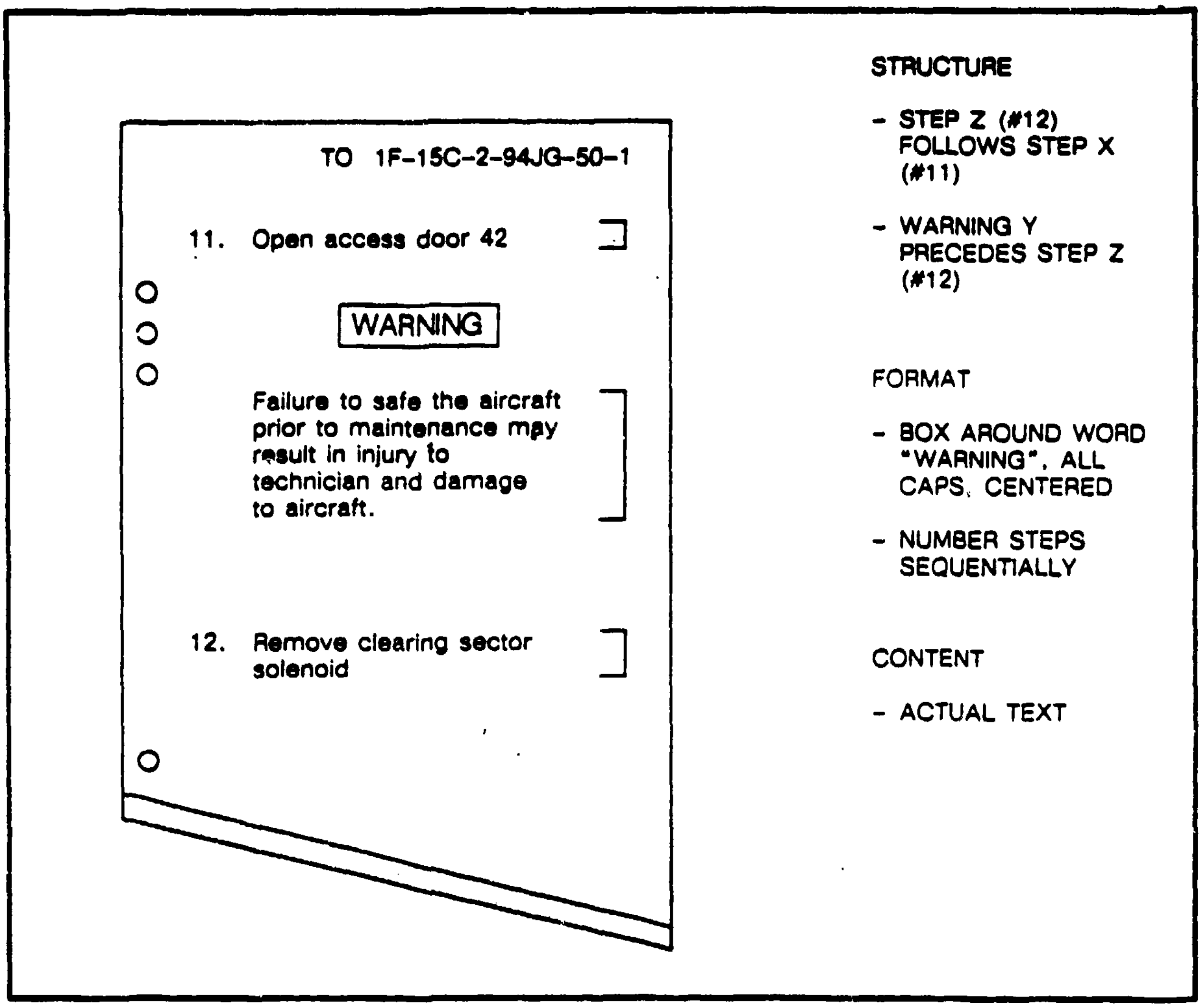
1. allow efficient interchange between contractors and the government of technical information.

2. allow quick and efficient presentation of the information on any current or future display medium.

3. eliminate redundant data, thus reducing data storage requirements.

4. allow rapid updating of the data as technical changes are implemented.

5 eliminate the need to change the data due to changes in the formatting requirements of the presentation system.



TO 1F-15C-2-94JG-50-1

11. Open access door 42

WARNING

Failure to safe the aircraft prior to maintenance may result in injury to technician and damage to aircraft.

12. Remove clearing sector solenoid

STRUCTURE

- STEP Z (#12) FOLLOWS STEP X (#11)
- WARNING Y PRECEDES STEP Z (#12)

FORMAT

- BOX AROUND WORD "WARNING", ALL CAPS, CENTERED
- NUMBER STEPS SEQUENTIALLY

CONTENT

- ACTUAL TEXT

Figure 4. Content, Structure, and Format.

Other Related Efforts

While AFHRL was conducting their research, other groups within the government and industry were developing their own strategies for the development and presentation of digital technical information.

Automated Technical Order System (ATOS)

The ATOS program was sponsored by the Air Force Logistics Command (AFLC). Its intent was to automate the capture, storage, and maintenance of page-oriented technical order data. It would also automate the distribution, management, and presentation of page-oriented technical order data.

ATOS gathered data from contractors and material management offices. ATOS also supplied technical orders for maintenance, repair, inspection, and modification functions. This provided for the productive coexistence of both the automated and manual technical order systems and allowed for the reduction and eventual elimination of manual technical order maintenance and distribution. The ATOS program was replaced by the Air Force Technical Order Management System (AFTOMS) program.

Improved Technical Data System (ITDS)

Improved Technical Data System (ITDS) is an interactive information delivery system developed by Northrop Corporation for the B-2 program. It supports the presentation of technical data to users engaged in the operation, maintenance, training, and support of equipment and systems. ITDS models technical data as a linear stream of information with Standard Generalized Markup Language (SGML) commands embedded within text and tabular data. The SGML markup codes identify the type of the technical data and specifies formatting instructions such as sizing, positioning, and color for electronic display or paper printing of the technical data. Through the use of SGML Document Type Definitions (DTDs), an application can rigorously define a class of documents such as job guides, flight manuals, and fault isolation procedures and prepare them for output.

MIL-M-28001

MIL-M-28001 is a military specification which defines the requirements for the digital textual data form of technical publications. Data prepared in conformance to the requirements in this specification facilitates automated preparation, storage, retrieval, exchange, and processing of technical documents. The requirements set forth by this specification include

1. procedures and symbols for markup of unformatted text in accordance with the military's specific application of SGML.
2. SGML compatible codes that will allow a technical publication to conform to specific content, structure and format requirements including MIL-M-38784 as applicable to technical manuals.
3. output controls that will provide a uniform structure for automated document processing functions.

Under the CALS initiative, MIL-STD-1840 specifies the delivery of digital technical information. The standard references a group of military specifications which specify the delivery requirements for certain types of data. Textual technical information must maintain compatibility with MIL-M-28001 by using SGML as the specification language. Graphical technical information developed on computer-aided design (CAD) systems must adhere to MIL-D-28000 or MIL-D-28003. Graphical technical information entered using electronic scanning equipment must conform to MIL-R-28002 (see Figure 5).

Standard Generalized Markup Language

SGML was used in the development of the CDM for several reasons: (a) CDM tagged data can be interchanged to the government and among contractors under MIL-M-28001, (b) SGML provides a checkable syntax so automated validation testing can be done, and (c) since storage of the data occurs in ASCII text files, each application of the data may use different data base management systems on different hardware systems and still be able to exchange and update data from other sources.

The International Standards Organization (ISO) developed SGML as an efficient way of transferring a wide range of documents between different types of computers. Used internationally, SGML breaks documents into text pieces using a standard set of markup

codes. The different markup codes describe the structure of the document. The SGML markup codes, also called "tags", are used to map the document. The SGML markup codes do not describe output formatting information.

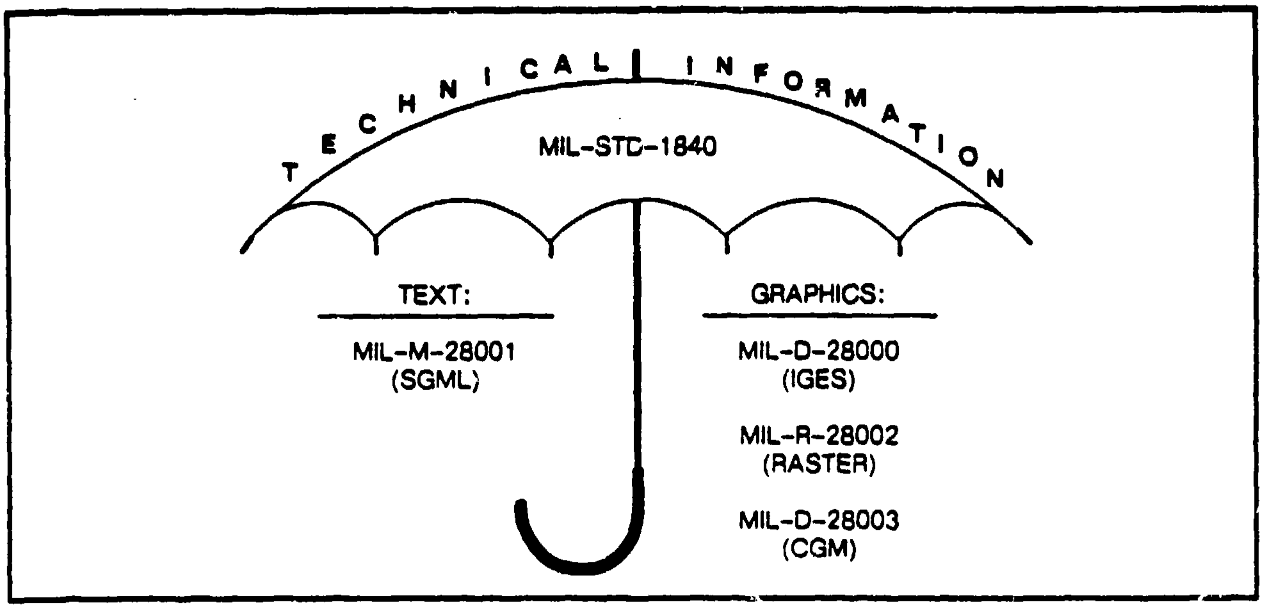


Figure 5. **MIL-STD-1840 and Associated Specifications.**

The SGML declaration defines syntax rules for the SGML tagging process. It defines

1. the actual characters used to distinguish tags from regular words within the document.

2. the limits for tag lengths.

3. whether tags will be in upper or lower case.

A common syntax which is applicable for most documents is defined in the ISO-8879-1986 standard for SGML.

The Document Type Definition (DTD) defines the grammar of the tag language used within a document. The grammar specifies the names of the tags being used and their possible position within a document. Each different type of document will probably have a different DTD. In order to further explain SGML, an inter-office memo will be used.

Some companies have policies for the information contained in a memo and the organization of a memo. For example, a memo may have a sender, a receiver, the date of the memo, and the subject of the memo. It would also contain paragraphs of text. A memo may also have attachments. A possible outline of the structure for the memo is shown in Figure 6.

I. MEMO

- A. Header
  - 1. Receiver
  - 2. Sender
  - 3. Date
  - 4. Subject
- B. Body
  - 1. Paragraphs
- C. Attachments

Figure 6. **Structure of a Sample Memorandum.**

The outline provides the structure of the memo. Using this structure the content may be entered as shown in Figure 7.

RECEIVER:	All Employees
SENDER:	John Doe
DATE:	June 7, 1987
SUBJECT:	Memorandum Formats (Policy 158-87)

PARAGRAPH:

In order to improve the flow of information throughout the company, a standard memorandum format is being adopted.

PARAGRAPH:

Every memo must have the heading which will include an addressee, a sender, a date, and the subject of the memo. The body of the memo will contain the actual text. Attachments may be made to the memo. If there are attachments, it must be noted on the memo.

Figure 7. **Content of a Sample Memorandum.**

In order to produce memos intended for the receivers, the memo needs rules which define the display of the information (see Figure 8). In this example, the memo will be printed on paper.

Receiver will be printed on the first line, left justified, preceded by the label, "TO:".

Date will be printed on the first line, right justified, with no label.

Sender will be printed on the third line, left justified, preceded by the label, "FROM".

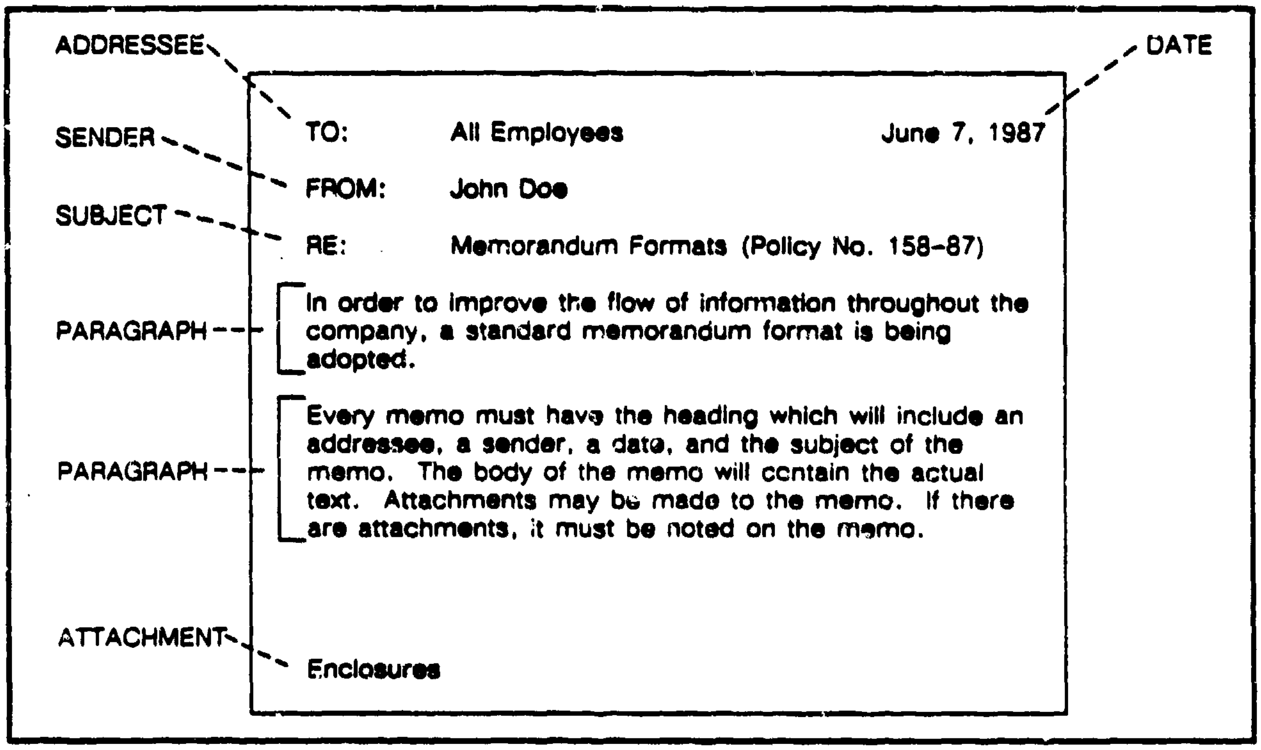
Subject will be printed on the fifth line, left justified, preceded by the label, "RE:".

Paragraphs will begin two lines below the last line printed. The first line of the paragraph will be indented five spaces.

If there are attachments to the memo, the word, "Enclosures" will be printed three lines below the last line printed.

Figure 8. Format Rules for a Sample Memorandum.

When the content of the memo is organized based on the structure and printed using the format rules, the output will appear as shown in Figure 9.



TO: All Employees June 7, 1987

FROM: John Doe

RE: Memorandum Formats (Policy No. 158-87)

In order to improve the flow of information throughout the company, a standard memorandum format is being adopted.

Every memo must have the heading which will include an addressee, a sender, a date, and the subject of the memo. The body of the memo will contain the actual text. Attachments may be made to the memo. If there are attachments, it must be noted on the memo.

Enclosures

Figure 9. Sample Memorandum.

Using the structure defined above, the SGML DTD for the memo may appear as shown in Figure 10.

```
<!DOCTYPE memo [
<!ENTITY  % text    "(#PCDATA)"                        >
<!ELEMENT memo       - -    (header, body, attach)   >
<!ELEMENT header     - -    (to, from, date, subject)>
<!ELEMENT to         - o    %text;                   >
<!ELEMENT from       - o    %text;                   >
<!ELEMENT date       - o    %text;                   >
<!ELEMENT subject    - o    %text;                   >
<!ELEMENT body       - -    (para)*                  >
<!ELEMENT para       - o    %text;                   >
<!ATTLIST para    number NUMBER   #REQUIRED          >
<!ELEMENT attach     - o    EMPTY                    >
<!ATTLIST attach  yesorno CDATA   'no'               >
]>
```

Figure 10. DTD for Sample Memorandum.

As mentioned before, the DTD defines the names of the tags which label the different data items. The DTD also defines the structure of the data.

A DTD consists of a document type (doctype), entities, elements, and attributes. The doctype simply gives a name to the DTD. In this example, "memo" is the name of the DTD.

An entity is an abbreviation or substitution mechanism. An entity declaration consists of a "%", a name for the entity, and a text string surrounded by quotes. Any time an entity occurs within the element or attribute declarations, the text is automatically inserted where the entity name occurs. For example, the line,

```
<!ELEMENT subject   - o    %text;         >
```

would look like,

```
<!ELEMENT subject   - o    (#PCDATA)      >
```

after the substitution. This allows the DTD developer to define items which may appear many times in the DTD as a single entity.

An element is a piece of data within a document. Within the element declaration, a name for the element is defined. Following the name, the DTD developer sets the requirements for beginning and ending tags. The beginning tag marks the beginning of a piece of data. For example, the beginning tag for a header would be entered as: <header>. The ending tag marks the end of a piece of data. A "-" signifies the requirement for an end tag. A "o" signifies that the tag is optional. For example, the ending tag for a header would be entered as: </header>. Next, the structure of the element is defined. An element may be made up of other elements, or it may contain actual data.

```
<!ELEMENT header    - -    (to, from, date, subject)>

<!ELEMENT to        - o    %text;                   >
```

In the example above, the "header" element must have beginning and ending tags. It is made up of "to," "from," "date," and "subject" information. The "to" element must have a beginning tag, but an ending tag is optional. The element definition includes the "text" entity which can be substituted for "#PCDATA" which is an SGML term for character data.

SGML also provides a mechanism for multiple occurrences of an element within another element. Occurrence indicators define these special cases. A "?" signifies that an element may occur zero or one time. A "*" signifies that an element may occur zero or many times. A "+" signifies that an element may occur one or many times.

```
<!ELEMENT body      - -    (para)*                  >
```

In the example above, there may be many "para" elements within a body, or there may be none, depending on the application.

Elements may have attributes associated with them. An attribute is a characteristic or property which does not contain actual content.

```
<!ELEMENT para      - o    %text;                   >

<!ATTLIST para      number NUMBER    #REQUIRED      >
```

In the example above, the "para" element also has an attribute called "number." This attribute defines a number each time the element is defined because the DTD designer has designated it as a required attribute.

Using the SGML markup codes described in the DTD, the user marks up or "tags" a document. An "instance" of a document is a marked up SGML document. The memo tagged using the DTD in Figure 10 is shown in Figure 11.

The tags define the content and the structure of the memo. For example, everything located between the "<header>" beginning tag and the "</header>" ending tag is header information. Also, notice the definition of the "number" attribute each time a "para" element was defined.

Computer programs, called parsers, check the DTD and the document instance. A DTD parser checks the DTD for correct and complete syntax. A document instance parser checks a tagged document for correct and complete tagging based on the document's DTD.

```
<memo>

<header>

<to> All Employees

<from> John Doe

<date> June 7, 1987

<subject> Memorandum Formats (Policy No. 158-87)

</header>

<body>

<para number = "1"> In order to improve the flow of information throughout the
company, a standard memorandum format is being adopted.

<para number = "2"> Every memo must have the heading which will include an
addressee, a sender, a date, and the subject of the memo. The body of the memo
will contain the actual text. Attachments may be made to the memo. If there are
attachments, it must be noted on the memo.

</body>

<attach yesorno="yes">

</memo>
```

Figure 11. Tagged Sample Memorandum.

III. CONTENT DATA MODEL

The CDM is the proposed model for an integrated weapon system data base intended to meet the long term needs of the DoD. CDM data emphasizes the identification of the content and the interrelationships (or structure) between data elements found within technical information.

CDM Concept

The CDM is a definition of interrelated technical information. This definition describes the technical information using its content and structure. Emphasis is not on the display of the data but on the data interrelationships and where the data fits together within a system. Many applications can use the data without modification. The CDM design does not concern itself with a particular output display. Since the information does not contain format, the information can be used with any output device such as paper systems, video displays and future technologies.

For example, a selected subset of CDM data can produce a particular manual and send it to an output device. Another subset of the data required to support interactive diagnostics for a particular subsystem can also be extracted and loaded into a device such as a portable flight-line computer. This will save money by cutting waste, reducing storage space, and minimizing updating requirements. .

The connections between elements identified within the CDM facilitate integration of the data. Data can be created in many ways and then loaded into the CDM. Once in CDM form, the content and structure of the data is preserved. Subsets of the CDM can then be extracted, converted, manipulated, and displayed on different devices to support a variety of objectives.

CDM Description

The development of the CDM has been an evolutionary process. Originally, the structure of the CDM followed the structure of the information found within the technical manuals. However, after initial demonstration and validation efforts, the structure was revised in order to more accurately model all of the technical information available for a weapon system.

The revised CDM attempted to more closely model the actual structure of technical information within the Air Force (see Appendix A). A hierarchy was established which described the weapon system and the corresponding information associatea with a specific element within the weapon system.

Within this hierarchy, a weapon system consists of smaller systems, which consist of subsystems, which consist of subassemblies, which finally consist of individual parts. The CDM models the technical information in this manner (see Figure 12). Information including tasks, descriptions, and trouble-shooting procedures connects to the weapon system at the level in which the information is used. For example, tasks and descriptive information which speak specifically about the M61 automatic gun system on an F-15 connect to the gun system while more general information may connect to weapons system or to the aircraft as a whole.

Although the CDM is modelled in SGML, the physical storage of the data base it defines could vary from application to application. Sequentially tagged ASCII data is suited for very few applications. However, the ASCII data can be input into any data base format by using the SGML tags to identify pieces of information. This is the reason SGML is a very suitable modelling language. SGML also had a very flexible structure which can be checked by computer programs that check to see if the SGML grammar is correct.. It allows for the identification of data elements to be separated from the formatted representations of those elements.

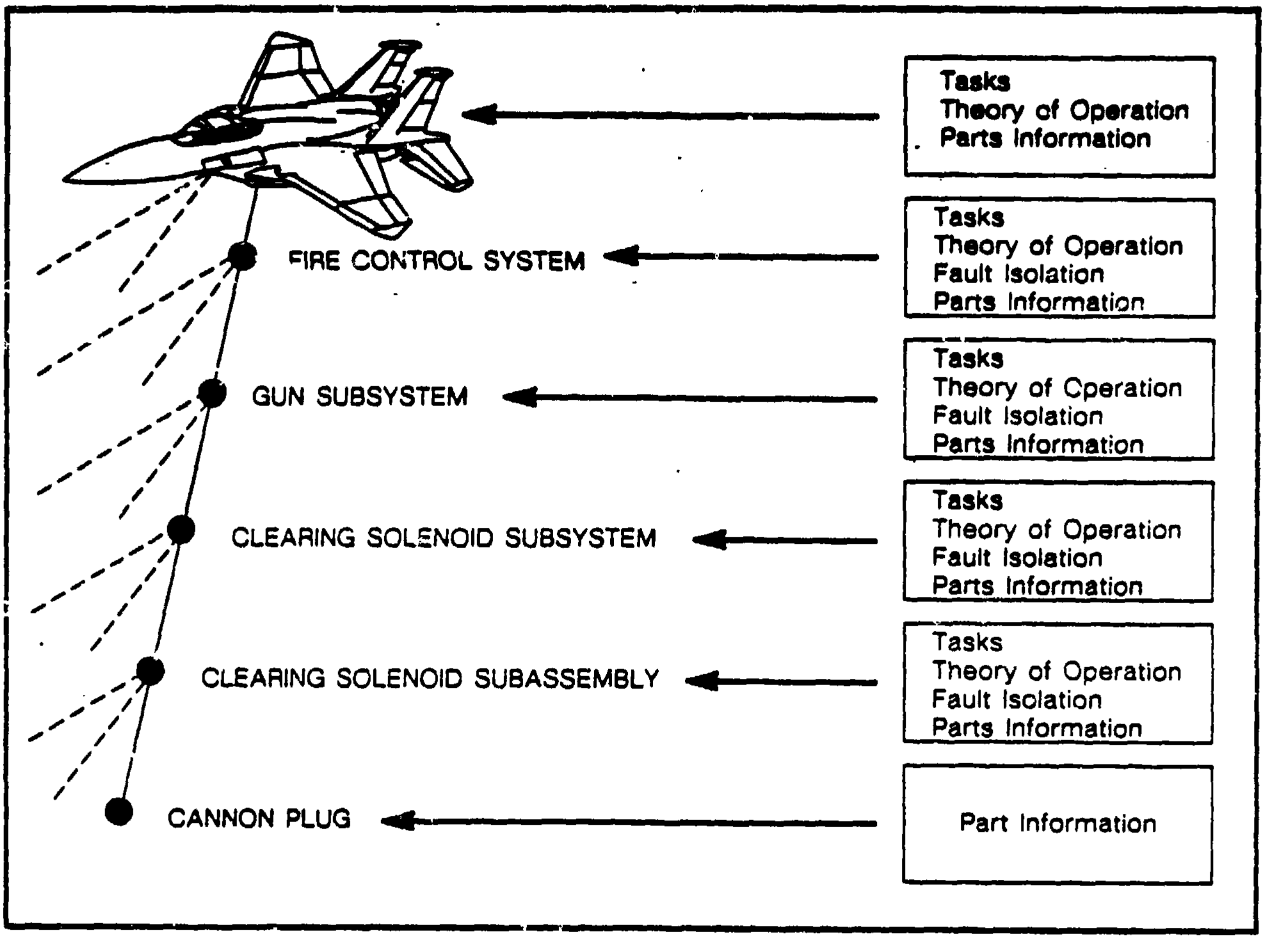


Figure 12. Weapon System Hierarchy.

CDM Elements

The main element of the CDM is "techinfo." This tag labels the beginning of the technical information. Within the techinfo structure, the first tag is the system tag. A system can represent the vehicle, an aircraft system, a subsystem, or a subassembly. At any level within this hierarchy, elements may reference associated information. These references may include procedural task information, descriptive information, parts information, fault information, or operational information. Figure 13 illustrates the CDM structure.

Procedural task information consists of maintenance tasks or procedures. A task consists of a set of input conditions, a list of steps, a list of follow-on tasks, and the estimated time and action for the task. Input conditions may include required conditions, personnel required, equipment required, and consumables. The input conditions and steps may also connect to warnings, cautions, notes, and user annotations which are comments entered by the technician as he is using the data.

Descriptive information defines general purpose, non-procedural, narrative information such as that contained in the theory of operation, schematics, and wiring diagrams.

There are two types of parts information. The first describes an item by its reference designator which arranges the parts by their place in the system-subsystem hierarchy.

This is the maintainer's view of the part information. This information is related to the second category of parts which describes the item by its part number. This is the supply system's view of the part information.

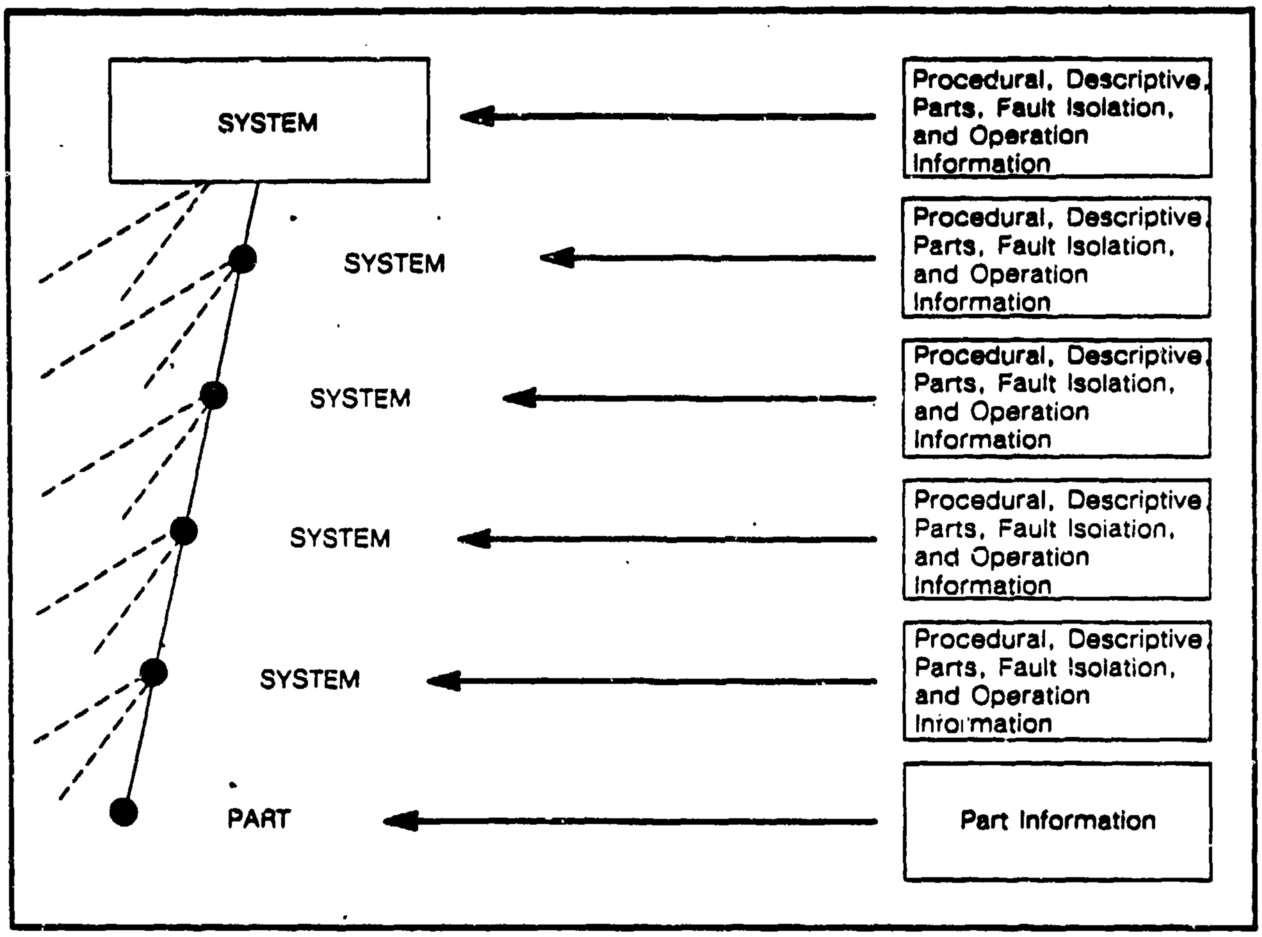

SYSTEM
Procedural, Descriptive, Parts, Fault Isolation, and Operation Information
SYSTEM
Procedural, Descriptive, Parts, Fault Isolation, and Operation Information
SYSTEM
Procedural, Descriptive, Parts, Fault Isolation, and Operation Information
SYSTEM
Procedural, Descriptive, Parts, Fault Isolation, and Operation Information
SYSTEM
Procedural, Descriptive, Parts, Fault Isolation, and Operation Information
PART
Part Information

Figure 13. CDM Element Hierarchy.

Three types of fault information can be described in the CDM:

- fault reporting decision trees;
- fault isolation decision trees; and,
- dynamic fault isolation model data.

The fault reporting and fault isolation decision trees are static, pre-defined decision sequences. Complete decision trees are defined in the data. A dynamic fault model generates the decision sequence at display time based on a fault model of the equipment being repaired. For a dynamic fault isolation model, only the data needed to represent the fault model of the equipment is defined in the data.

All of the diagnostic structures consist of diagnostic tests, test outcomes, fault states, repairable faults, and fault rectification actions. The fault reporting and isolation process begins with a diagnostic test which will have associated outcomes. These outcomes connect the test outcomes with new fault states. Test outcomes are described as preconditions which are asserted by the test procedure as the test is being performed. The test outcome element relates the possible test outcomes to fault states. Once a fault is identified, the rectification procedure associated with the fault is performed.

The technical information in the CDM consists of several types of data. These include text, graphics, audio sequences, video sequences, and external software processes. Elements within the CDM can be cross-referenced to other elements within the CDM or to external information. Prompts define one type of user interaction. A prompt may be either a fill-in-the-blank or menu choice for the user. Prompts assert properties depending on the user's response to the prompt. A property's value can be asserted or tested by the presentation software. An assertion is made whenever a task or step is performed. A precondition is a test for the status of a property. It is also possible to define a prompt or task to acquire a value for a property if the value does not already exist.

The CDM allows access of technical information in a hypertext-like manner. Hypertext is a system which allows the user to access related data in a data base from any specific point within the data base. It also allows the presentation of different information depending on the circumstances. CDM elements may have "context" which determine the applicability of a particular data element to the situation at hand. Properties, preconditions, and assertions may also define the conditions at any given point in time.

CDM Attributes

SGML attributes define the relations between the elements. Two common attributes include the "id" and "refid" attributes. These two attributes are used as element identifiers. These identifiers are machine generated symbols used to identify the data elements. These identifiers are not human readable names but only machine readable pointers or references. Most elements within the CDM have both the unique identifier (id) and the non-unique reference identifier (refid). The reference identifier is used by the elements to refer or point to other elements. For example, a task element may refer to a list of steps. Since the reference identifier is not unique, it may refer to several elements within the data base. These referenced elements will have a different context which will be used to determine which unique element is appropriate for a particular situation.

When one element refers to another element, the first element has an attribute which is named identically to the second element. The attribute value is a list of reference identifiers pointing to the appropriate elements. For example, if a task refers to a set of steps, the step attribute will have a list of reference identifiers which point to the appropriate groups of steps as an attribute value.

Other common attributes include the name, type, and item identifier attributes. The name attribute designates the name of the element's content. The type attribute designates the type of the element. The item identifier (itemid) identifies the equipment using a specific identifier. For example, a system might have a name, "GUN SYSTEM", a type, "M61A1" and an item identifier which may be the system, subsystem, subject number (SSSN) of a system or tail number of an aircraft.

IV. CDM VALIDATION EFFORTS

After the Content Data Model had been developed, it was necessary to prove that the model could accomplish the planned objectives and produce the expected benefits in a practical, cost effective manner. To demonstrate this, sample data was developed using

CDM requirements. The data was then output in several formats to demonstrate the flexibility of the model.

Objective

The objectives of the CDM validation efforts were to demonstrate that: a) the CDM could efficiently produce and present technical information for a weapon system in today's and tomorrow's maintenance environment, and b) that an integrated data base could produce the expected benefits.

Approach

A three phased approach was implemented to validate the CDM. The three phases included concept exploration, limited demonstration and validation, and full-scale demonstration and validation. This approach closely resembles the Air Force's acquisition approach for a system. In order to validate the integrated data base concepts, a data base using the CDM structure was developed and data loaded into and extracted from it in different forms.

Concept exploration began with the examination of current technical manuals. This examination provided information about the types of data required and the structure of the data within the technical manuals. It also provided insight into ways of optimizing the data for an integrated data base. A portion of the exploration included a study to determine the compatibility of CDM data with ITDS. A compatibility study was also performed for APS. These comparisons provided additional information concerning the content and structure of technical information. The information from these studies aided in the development of the CDM.

In the limited demonstration and validation phase, the CDM structure was reviewed. This structure resembled the structures found within the technical manuals. A small amount of data contained in paper manuals was entered into a CDM data base to determine if the structure was appropriate for the intended information. Software routines converted ITDS data to CDM and vice versa. Another routine converted ITDS to gencodes used by the ITDS data delivery system. Additional conversion routines were developed to convert APS data to CDM and vice versa.

After the demonstration and validation phase, the CDM structure was modified to more accurately model the data. The new structure modelled the information hierarchically, based on the structure within a specific weapon system. This model was used for the full scale demonstration and validation phase. In the full-scale demonstration and validation phase, a complete set of information was entered into the CDM for a specific subsystem of a major weapon system. Data was entered into and extracted from the CDM in a number of ways during both demonstration and validation phases, validating the concept of an integrated data base.

Documentation produced during the effort was developed to provide a means in which to procure integrated CDM technical information. This documentation included a data exchange standard, a data item description, and an implementation specification for digital technical information.

Compatibility Studies

A major consideration in the exploration of the CDM concept was the requirement that CDM data be compatible with data developed and stored in other formats. This upward compatibility gives the Air Force the ability to upgrade technical information to the CDM format without the need to re-develop the data. To assure compatibility with other technical data base systems either currently in use or in development, the compatibility of data between the CDM and various existing systems was studied. Systems which were examined included:

1. Digital data intended for paper presentation developed under MIL-M-28001.
2. ITDS, an electronic data delivery system not already in use.
3. APS, an electronic authoring and data delivery system developed by AFHRL.

Compatibility with MIL-M-28001

Data developed for delivery on paper was included to determine if the CDM could represent existing technical information. The ability to convert existing technical information allows the Air Force to convert existing digital data for weapon system manuals into the CDM.

MIL-M-28001 contains two DTDs in its appendices. One DTD defines the structure for technical manuals developed under the requirements set forth in MIL-M-38784. The other DTD defines a structure for documents not covered by MIL-M-38784.

Method. The SGML tags and structures in the MIL-M-38784 compliant DTD contained within MIL-M-28001 were examined for compatibility with the CDM. Rules for the conversion of the data were defined which allowed the data to retain its content and structure while deleting any format information which was connected to it.

Results. It was determined that data tagged under the MIL-M-38784 compliant DTD could be loaded into a data base developed using the CDM structure.

Compatibility with ITDS

The ITDS technical data system was examined for compatibility with the CDM. ITDS implements SGML in a different manner than the CDM.

Method. A study was done to determine the feasibility of converting data between ITDS and the CDM. The discovery of some minor incompatibilities led to recommendations for design changes to both systems which would make them compatible.

Results. After the implementation of the changes, it was concluded that although fundamental differences in the implementation techniques of the two systems remained, the two systems were compatible. These remaining differences would not prohibit the exchange of information between the two systems.

The main differences which now exist between ITDS and CDM involve their implementation methodologies. ITDS models the technical data sequentially using SGML commands embedded within the data. ITDS stores format information within the text, causing a need to store a piece of data multiple times. The CDM models maintenance information hierarchically and uses SGML markup codes to identify information content and structure. The CDM does not retain any format information allowing the same piece of data to be stored once and used many times.

Compatibility with APS

CDM was also tested for compatibility with the APS for several reasons. First, APS data is stored in a relational data base structure with the data distributed in many files.

Also, the interrelationships between the elements of the data base closely resemble those within the CDM. Finally, because it was developed by AFHRL, it was readily available. The organization of the APS data base would be a test to validate the assumption that data stored in different formats was convertible to CDM data.

Method. Since APS was a direct predecessor to the CDM, a complete compatibility study was not done. However, the data elements were examined and compared to those in the CDM. The interrelationships between the elements were also examined.

Results. It was concluded that data developed using APS could be loaded into a data base developed according to the CDM structure.

Results of Compatibility Studies

MIL-M-28001, ITDS, and APS data were all found to be compatible with the CDM. The next step was to develop a CDM data base and populate it with actual technical information. In addition to the development of the data base, conversion routines would be developed to demonstrate that data could be interchanged between the different systems.

Limited Demonstration/Validation

After the determination was made that the CDM was compatible with several existing standards and systems, the concept had to be demonstrated and validated. A small sample of data was used to demonstrate that the various data structures contained in the technical information could be modelled in the CDM.

Data Selection

In order to validate the CDM concept, CDM data was developed. The use of existing maintenance information allowed the current technical manual requirements to be modelled as closely as possible. The data used in the demonstration and validation phase was initially limited to organizational level maintenance manuals for the M61A1 and M61 automatic gun system as implemented on the F-15 and F-16 aircraft. The gun system was chosen to demonstrate that the same data could be used on many different weapon systems. It was thought that if the gun system was not altered from one aircraft to another, the CDM for the F-15 gun system would be very similar to the CDM developed for the F-16 gun system.

During this investigation, it was discovered that the CDM data bases for the F-15 and the F-16 were not the same. The presentation of the technical information within the manuals was different for each weapon system even though the same gun was being used. In some cases, the information sequence was also different. The investigation determined that the differences were subtle and would not alter the test.

Five job guide tasks and five fault isolation procedures were chosen for the F-15. Corresponding tasks and procedures were chosen for the F-16.

Data Mapping

This phase included the development of DTDs for the job guide and fault isolation manual.

Job Guide DTD. The development of the job guide DTD used the MIL-M-38784 compliant DTD contained in MIL-M-28001 as a baseline. Additional structures were developed by examining and mapping the data required for job guides in accordance with MIL-M-83495. Existing job guides were also examined to determine any necessary additions or revisions in order to correctly model current job guides. Several differences were discovered between the F-15 and F-16 job guides; however, most of these differences involved the format of the manuals which was not included in the DTD. Appendix B contains the job guide DTD.

Fault Isolation Manual DTD. The development of the fault isolation manual DTD used the MIL-M-38784 compliant DTD contained in MIL-M-28001 as a base line. The Fault Isolation Manual DTD was developed because of the structural differences between the two manuals. MIL-M-83495 was used to develop some of the structures while examination of existing fault isolation manuals provided other structural information. As was the case with the job guides, some format differences were found between manuals from the two weapon systems. These format differences had no bearing on the DTD. Appendix C contains the fault isolation manual DTD.

Data Entry

The textual data contained in the job guides and fault isolation manuals was entered into electronic format using conversion routines and direct entry. A scanner, which produced raster images, was used to enter graphic data. A CAD program, which produced vector images, was also used. Figure 14 illustrates the processes used in the creation and entry of the data.

Conversion. The textual data for the F-16 job guides and fault isolation manuals was originally provided in electronic form on magnetic tape. The data was loaded onto a Digital Equipment Corporation VAX computer. The data was then downloaded from the VAX to an IBM PC computer using communications software. Once the data had been downloaded, random characters remained within the file. A pre-processing computer program removed these characters from the file.

Direct Entry. Data for the F-15 was manually entered using a word processing program on the IBM PC since the data was not readily available in electronic form. Two different strategies were used. In the first, the text was entered followed by the insertion of SGML tags. In the second, the text was tagged as it was entered.

Tagging. Tagging of the F-15 and the F-16 data was initially done using the Writerstation, a tagging program developed by Datalogics. The Writerstation used an ASCII text file containing the DTD for the document to be tagged and the text file containing the untagged data as input. The program used the DTD to prompt the user for the appropriate tag and attribute information which was then inserted into the text to produce an SGML document instance. Due to the numerous changes in the DTDs as the data was being developed and tagged, the Writerstation was abandoned in favor of an ASCII text editor. Using the text editor, tags and attributes were inserted in the appropriate places within the text file.

Parsing. Following the tagging of a document instance, a parser verified conformance to the appropriate DTD and to the SGML standard. The parser program checked for correctness of the markup as well as tag sequence and completeness. If errors were found, they were corrected and the file was reparsed until no more errors occurred.

Data Loading

The tagged document instances for each type of manual was converted from ASCII text files into relational data base structures which were designed based upon the CDM structure. The information was loaded into Canonical Document Representation (CANDOR) relational data base tables (1 table per manual type) which allowed random access to the data. The process involved reading the document instance for each document and inserting the data into the appropriate location within a CDM data base. This process produced a CDM data base for each document type. The manual combination of the individual CDM data bases produced a single, complete CDM data base. Figure 15 illustrates the conversion of the technical manual data.

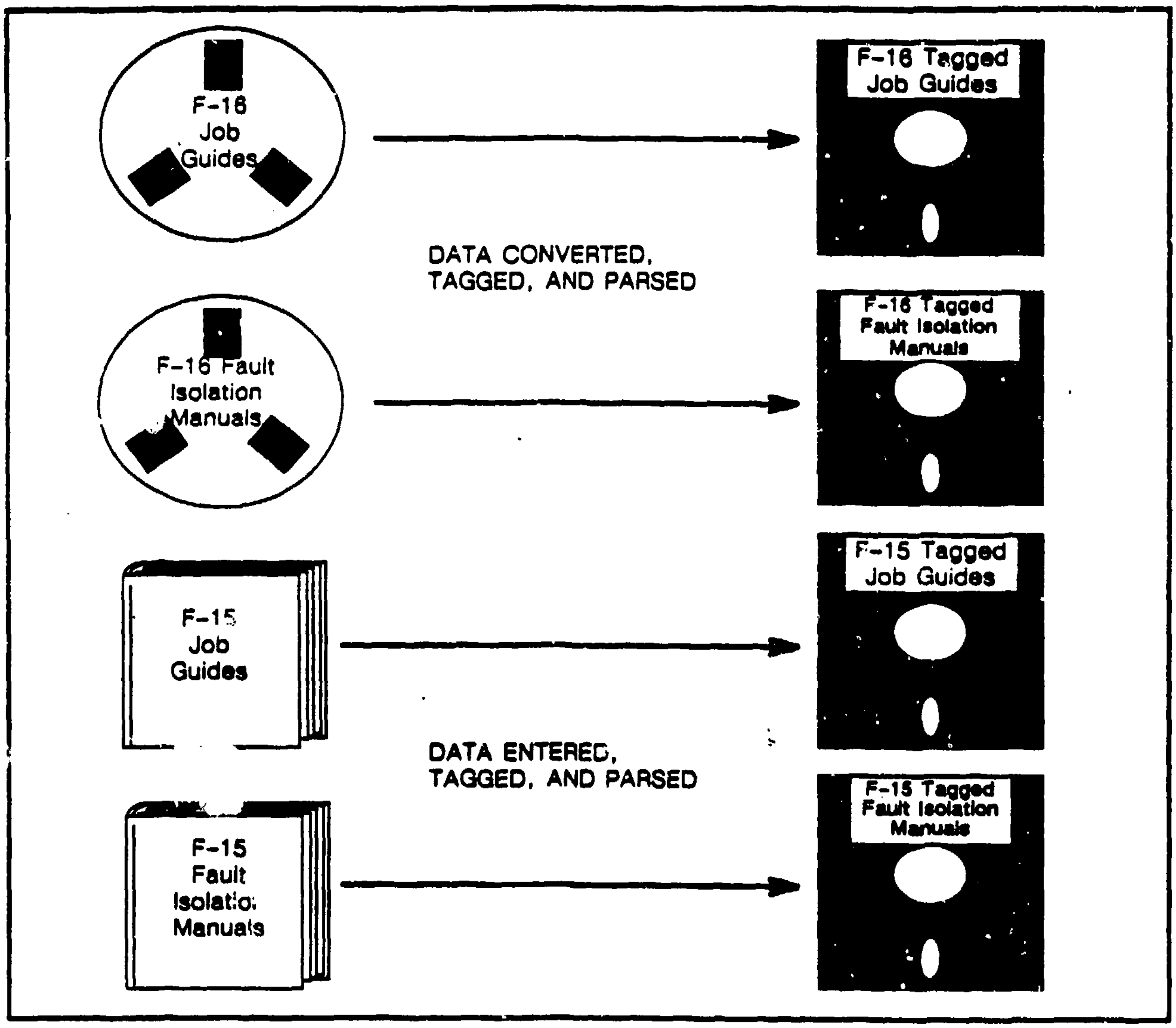

F-16 Job Guides
F-16 Tagged Job Guides
DATA CONVERTED, TAGGED, AND PARSED
F-16 Fault Isolation Manuals
F-16 Tagged Fault Isolation Manuals
F-15 Job Guides
F-15 Tagged Job Guides
DATA ENTERED, TAGGED, AND PARSED
F-15 Fault Isolation Manuals
F-15 Tagged Fault Isolation Manuals

Figure 14. Data Entry For Limited Demonstration/Validation.

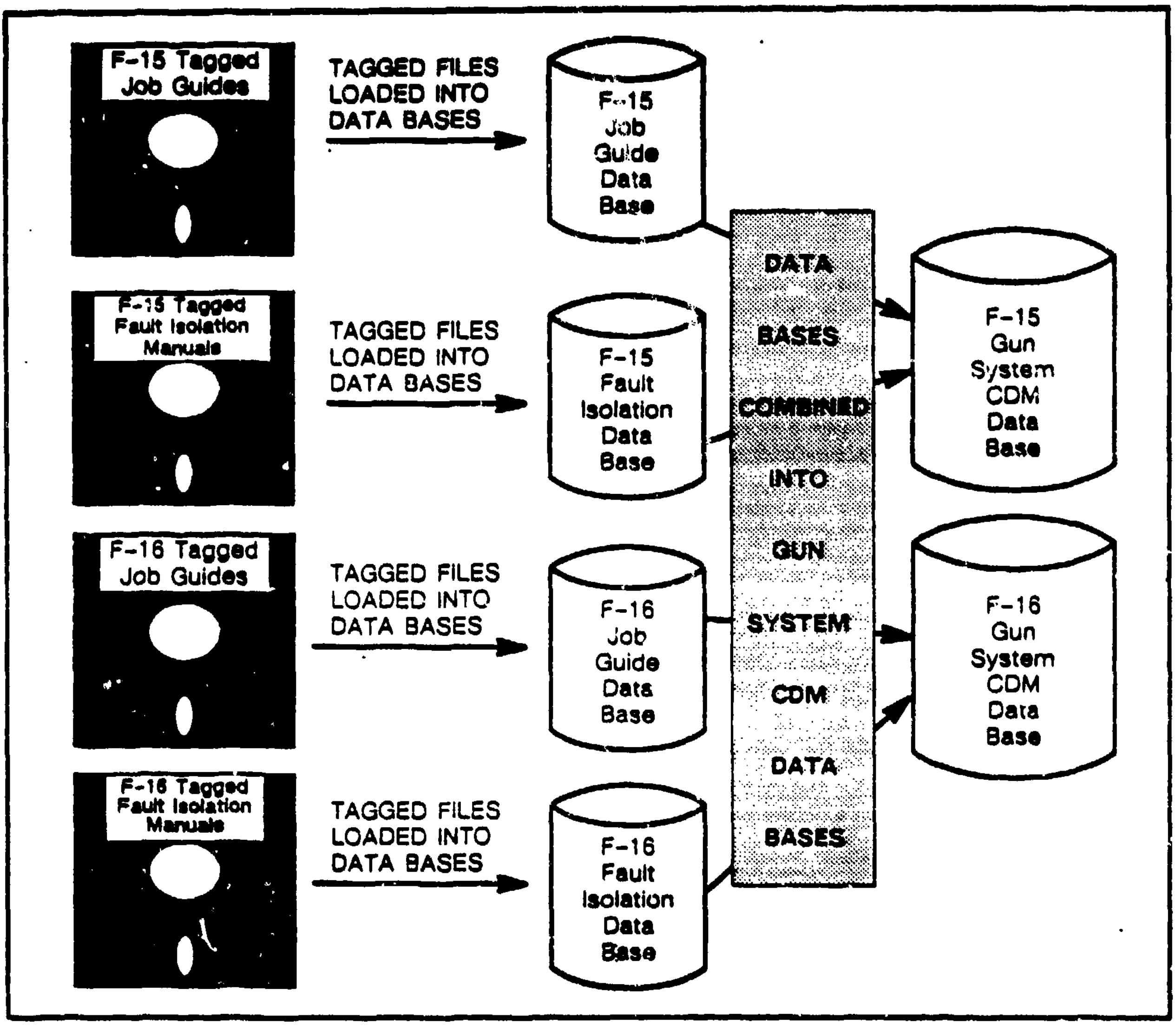

F-15 Tagged Job Guides
TAGGED FILES LOADED INTO DATA BASES
F-15 Job Guide Data Base
F-15 Tagged Fault Isolation Manuals
TAGGED FILES LOADED INTO DATA BASES
F-15 Fault Isolation Data Base
F-16 Tagged Job Guides
TAGGED FILES LOADED INTO DATA BASES
F-16 Job Guide Data Base
F-16 Tagged Fault Isolation Manuals
TAGGED FILES LOADED INTO DATA BASES
F-16 Fault Isolation Data Base
DATA BASES COMBINED INTO GUN SYSTEM CDM DATA BASES
F-15 Gun System CDM Data Base
F-16 Gun System CDM Data Base

Figure 15. Data Loading for Limited Demonstration/Validation.

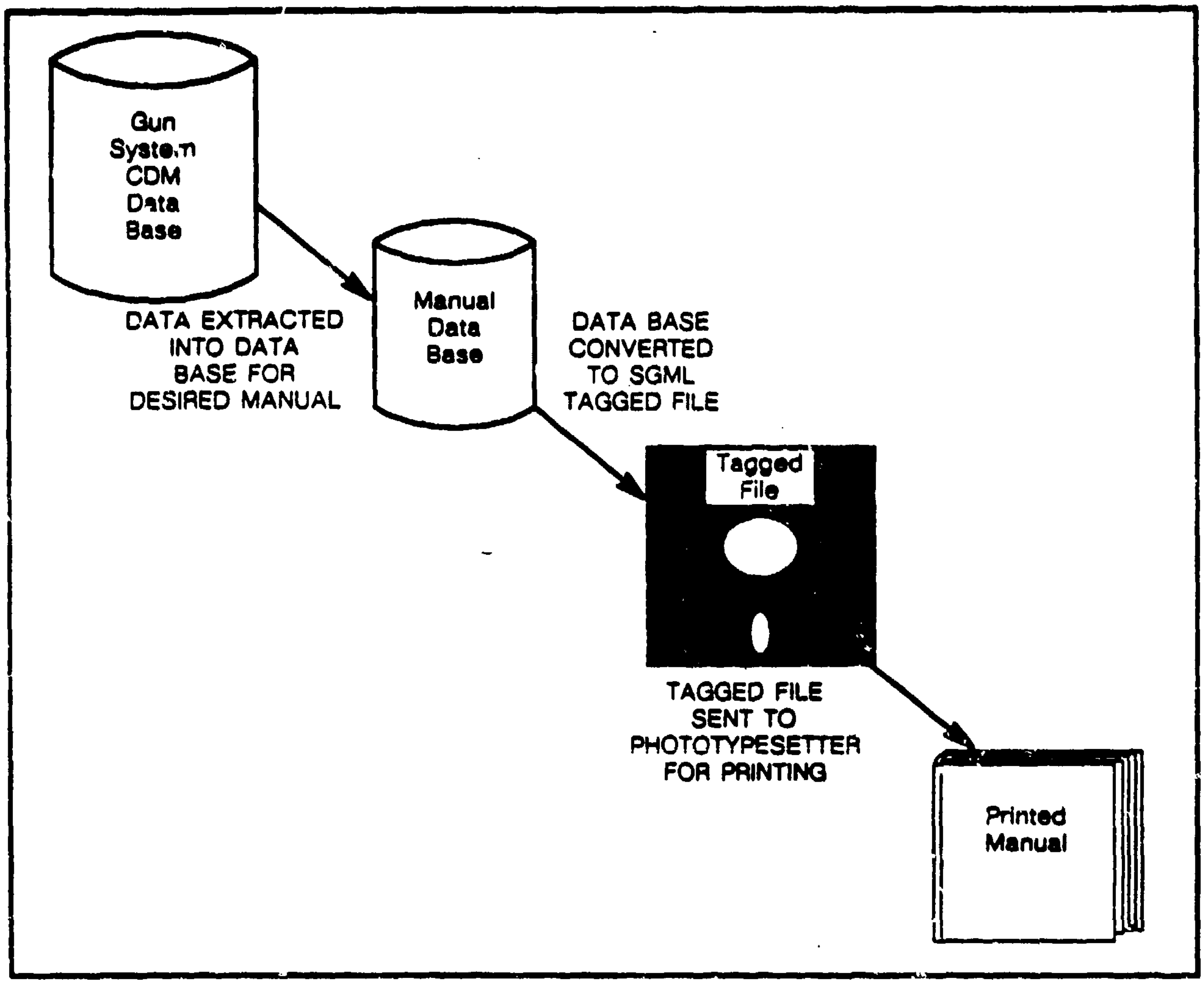
Data Extraction

The data extraction process involved a reverse of the loading process. A program, which extracted the necessary data for the particular manual, read the data stored in the CDM data base. The extracted data was then assembled into a separate data base before conversion into a SGML-tagged document instance. Figure 16 illustrates the processes used for the extraction of data from the CDM.

In order to perform a quality check on the extracted information, a phototypesetter routine was used to produce paper output. Since the original data being loaded into the CDM was primarily from paper manuals, paper output allowed for easy comparison of the original data and the extracted data.

In order to typeset the manuals, an interim output specification was developed. This output specification provides rules for the formatting of the data in approximately the same way as the original manuals. During the process of developing the output

algcrithms, differences in format were found between the manuals for the different weapon systems. In cases where differences were found, generalizations were made for the output specificaticn.



Gun System CDM Data Base
DATA EXTRACTED INTO DATA BASE FOR DESIRED MANUAL
Manual Data Base
DATA BASE CONVERTED TO SGML TAGGED FILE
Tagged File
TAGGED FILE SENT TO PHOTOTYPESETTER FOR PRINTING
Printed Manual

Figure 16. Data Extraction for Limited Demonstration/Validation.

Data Conversion

After the development and testing of the data loading and extraction procedures, routines were developed which converted data which was developed and stored in different formats into the CDM format. These conversions demonstrated that data developed and used on other systems was transferable to and from the CDM without the need to revise the original data.

Data was converted between ITDS and CDM. To do this, ITDS job guide-like data was converted into a job guide document instance. Since paper job guide data was transferable to the CDM, it was assumed that ITDS data converted to job guide DTD format as a preliminary step, could be loaded into the CDM.

ITDS to CDM. The ITDS system used its own hardware to deliver the data, both electronically and on paper. This required the inclusion of formatting commands. Since the CDM retains only the content and structure of the data, the formatting commands were automatically removed as the conversion was done. Many of the elements within the job guide DTD also required the use of SGML end tags to denote the end of a segment of data. These tags were not required in the ITDS scheme. A two-phase conversion process accomplished the conversion of the data and the insertion of the end tags. Figure 17 illustrates the two phases used in the conversion from ITDS to CDM.

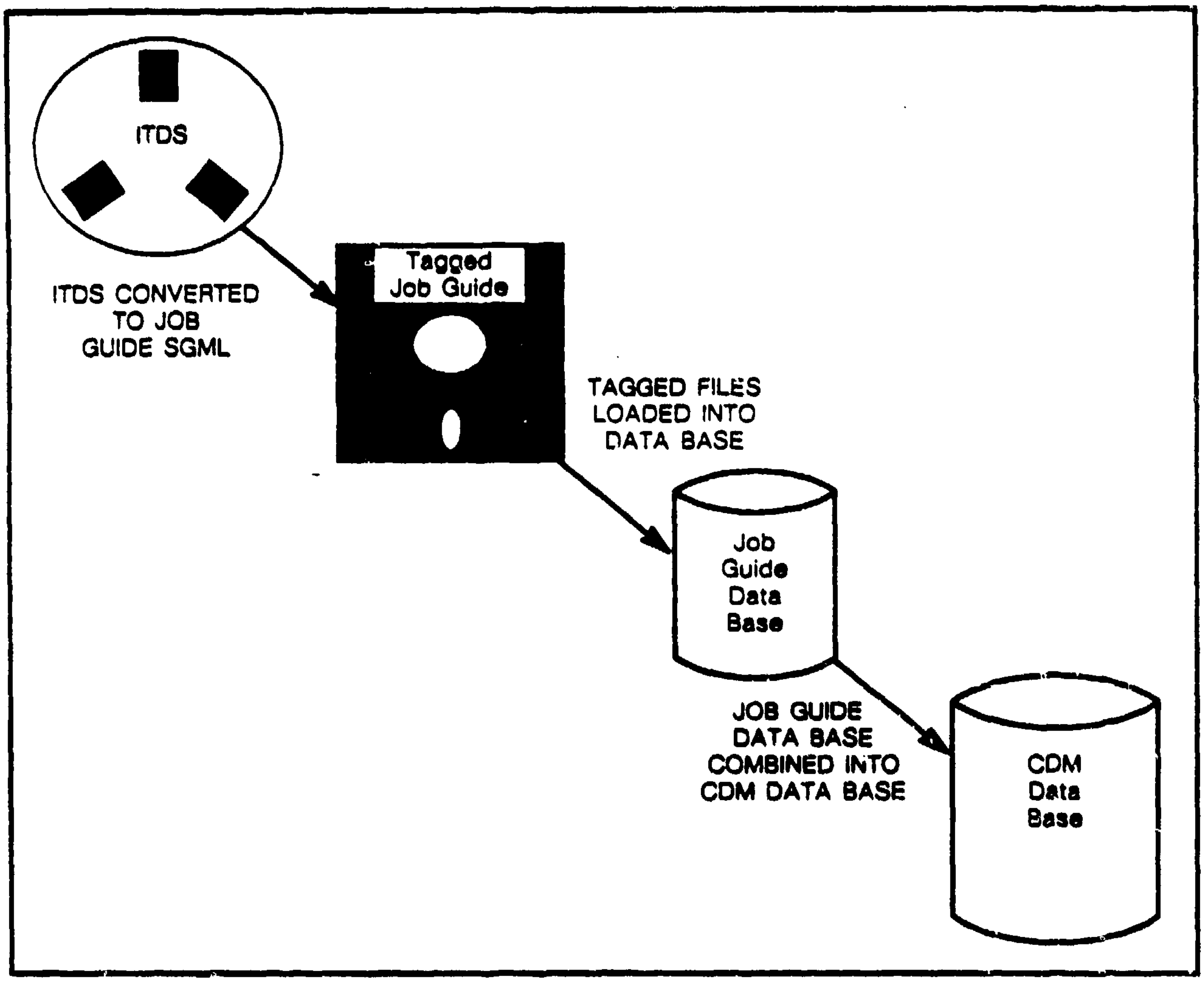

ITDS
ITDS CONVERTED TO JOB GUIDE SGML
Tagged Job Guide
TAGGED FILES LOADED INTO DATA BASE
Job Guide Data Base
JOB GUIDE DATA BASE COMBINED INTO CDM DATA BASE
CDM Data Base

Figure 17. ITDS to CDM Conversion.

The first phase involved the conversion of the ITDS data to the appropriate SGML tags and then the removal of formatting information. A program read the input file containing ITDS data. Once a tag was found, the program referenced a tag dictionary which contained all of the job guide tag equivalents. If an equivalent tag existed, the program removed the ITDS tag and inserted the equivalent job guide tag. If an equivalent tag was not available, it was assumed that the tag was a format tag and thus, not required in the CDM. Following the tag exchange, the job guide tag and actual data were written to a temporary output file. This process continued until the end of the input file.

The second phase involved the insertion of the end tags in the proper places within the output file. A hierarchy of the tags contained in the job guide DTD was developed. A subroutine read the temporary file. Once a tag was encountered the program recorded the location of the tag within the hierarchy. If a tag of lesser value was encountered, the program continued to search for the next tag. If a tag of equal value was encountered, the program inserted an end tag into the output file for the previous tag at that level, recorded the new tag, and continued to search for the next tag. If a tag of higher value was encountered, the program inserted end tags into the output file for all the tags up to the level of the encountered tag, recorded the new tag, and continued to search for the next tag. This process continued until the end of the temporary file. At that point, the program closed all of the files and stopped.

After the completion of the conversion process, a parser checked the generated job guide document instance to determine adherence to the job guide DTD and the SGML standard.

CDM to ITDS. The conversion of data from the CDM to ITDS required the insertion of formatting tags. ITDS requires these formatting tags, but they are not included in the CDM. In this conversion, job guide data was extracted from the CDM and stored as job guide document instances. A program read the input file containing the job guide data. Once a tag was found, the program referenced a tag dictionary containing all of the ITDS tag equivalents. If an equivalent tag existed, the program removed the job guide tag and inserted the equivalent ITDS tag. If an equivalent tag was not available, it was assumed that the tag was not required in ITDS, so no equivalent tag was required. After the tag exchange, the ITDS tag and actual data were written to an output file. This process continued until the end of the input file was reached. At that point the program closed all of the files and stopped. Figure 18 illustrates the CDM to ITDS conversion process.

During this conversion, formatting commands were automatically inserted into the file as the tags were converted. Since the CDM does not store format, the routine had to make some formatting assumptions which would typically be made by the ITDS technical writer. These assumptions were implemented by using the "graphic" tag used in the job guide DTD to trigger the insertion of format tags. Job guides are printed with instructions on the left page and the accompanying graphic on the facing page on the right. Whenever a reference to a new graphic occurred, it was assumed that a page break would be needed in the ITDS paper output, or a different graphic would need to be displayed in the electronic output. At these points the appropriate format commands were inserted into the output file. These insertions formatted the data in approximately the same manner as a technical writer would format the data for use in ITDS.

ITDS to Gencode. At the time of this effort, machines running under the ITDS tagging structure used Document Exchange Standard codes or gencodes to tag the data. Gencodes are a more cryptic tagging scheme which were adopted before the SGML standard was developed. In order for data tagged using ITDS SGML tags to be used on an ITDS machine, it had to be translated into the gencode format. A program read an input file containing ITDS SGML data. Once an ITDS SGML tag was found, the program referenced a tag dictionary containing all of the gencode tag equivalents. If an equivalent tag existed, the program removed the ITDS SGML and inserted the equivalent gencode tag. Following the tag exchange, the gencode tag and actual data were written to an output file. This process continued until the end of the input file at which time the program closed all of the files and stopped. Figure 19 illustrates the process used to convert ITDS data to gencode tagged data.

An operational ITDS machine was not available for use in testing the data. Instead, Northrop ITDS technical writers were consulted for their opinion of the quality of the gencode-tagged data. It was the general consensus that the data conversion was successful and the data would be compatible with ITDS.

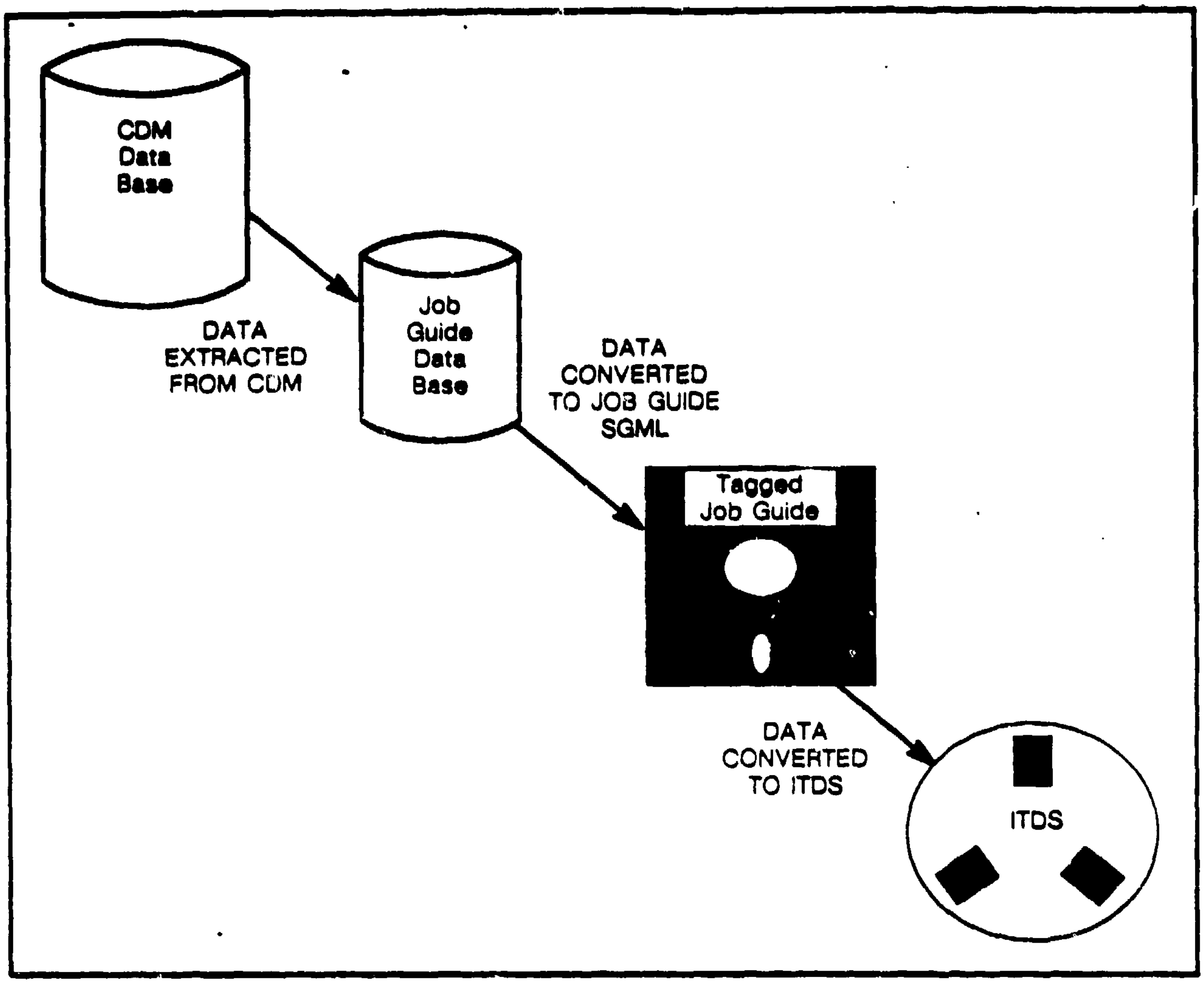


Figure 18. CDM to ITDS Conversion.

A conversion from gencodes to ITDS SGML was not included since changes to the SGML standard had not been finalized by the National Institute of Standards and Technology (NIST) and the Aerospace Industries Association (AIA).

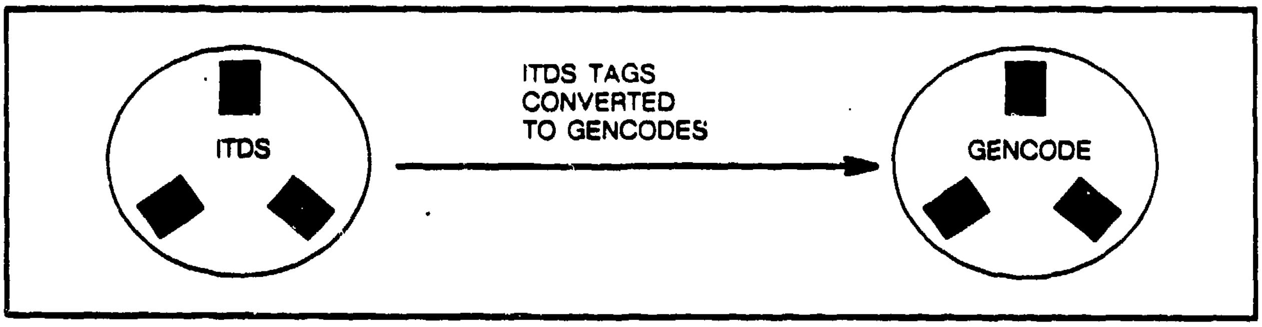


Figure 19. ITDS to Gencode Conversion.

To demonstrate that data developed using a separate authoring system and stored in a different data base layout could be compatible with the CDM, a conversion of APS data to CDM data was accomplished by AFHRL in-house resources and RJO.

APS to CDM. Data developed using APS was stored in a relational data base in binary format. In order to load this data into the CDM, an APS DTD was developed. Next, a program converted the APS binary information into an APS document instance. The program opened a configuration file which contains the order in which the thirteen APS binary files are to read. This file acts as a control mechanism for the program. As the program read each input file, the element and attribute information is extracted and converted into an SGML format. The SGML information is then written to the output file. This continues until all of the input files have been converted at which time the program closes all of the files and stops. Once the APS SGML file was completed, the data could then be loaded into the CDM CANDOR tables. Figure 20 illustrates the conversion of APS data to the CDM.

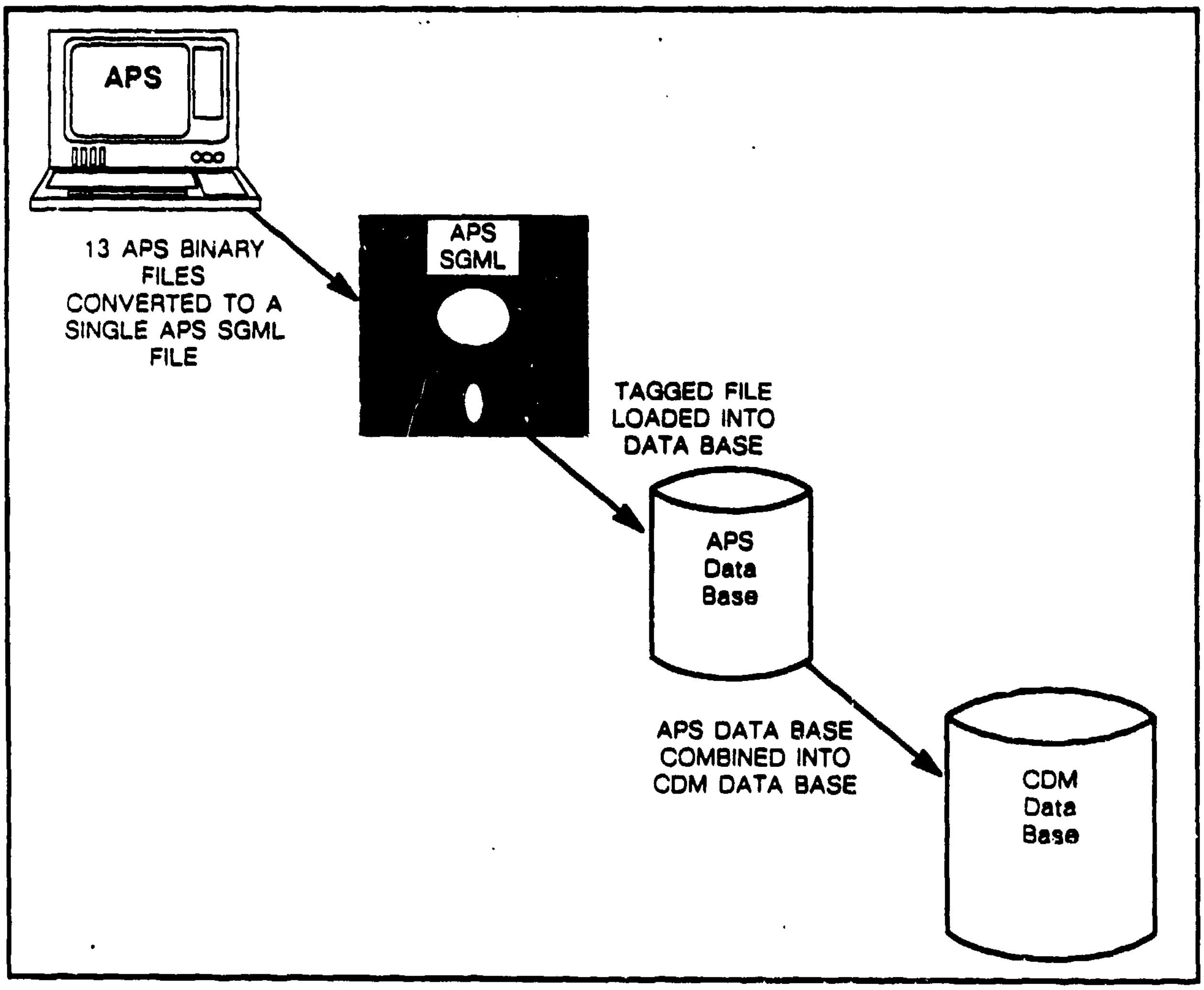

APS
13 APS BINARY FILES CONVERTED TO A SINGLE APS SGML FILE
APS SGML
TAGGED FILE LOADED INTO DATA BASE
APS Data Base
APS DATA BASE COMBINED INTO CDM DATA BASE
CDM Data Base

Figure 20. APS to CDM Conversion.

CDM to APS. The conversion of CDM data to APS binary data involved three steps. First, the required data was extracted from the CDM CANDOR table and written to an APS SGML text file. In the second step, a program read the APS SGML text file for

SGML tags. As tags were encountered, attribute data was written to temporary ASCII files depending on the tag encountered. After the extraction of all of the attribute information, the program read and converted temporary files into a binary data base fonnat. Figure 21 presents the steps taken to convert CDM data back to APS.

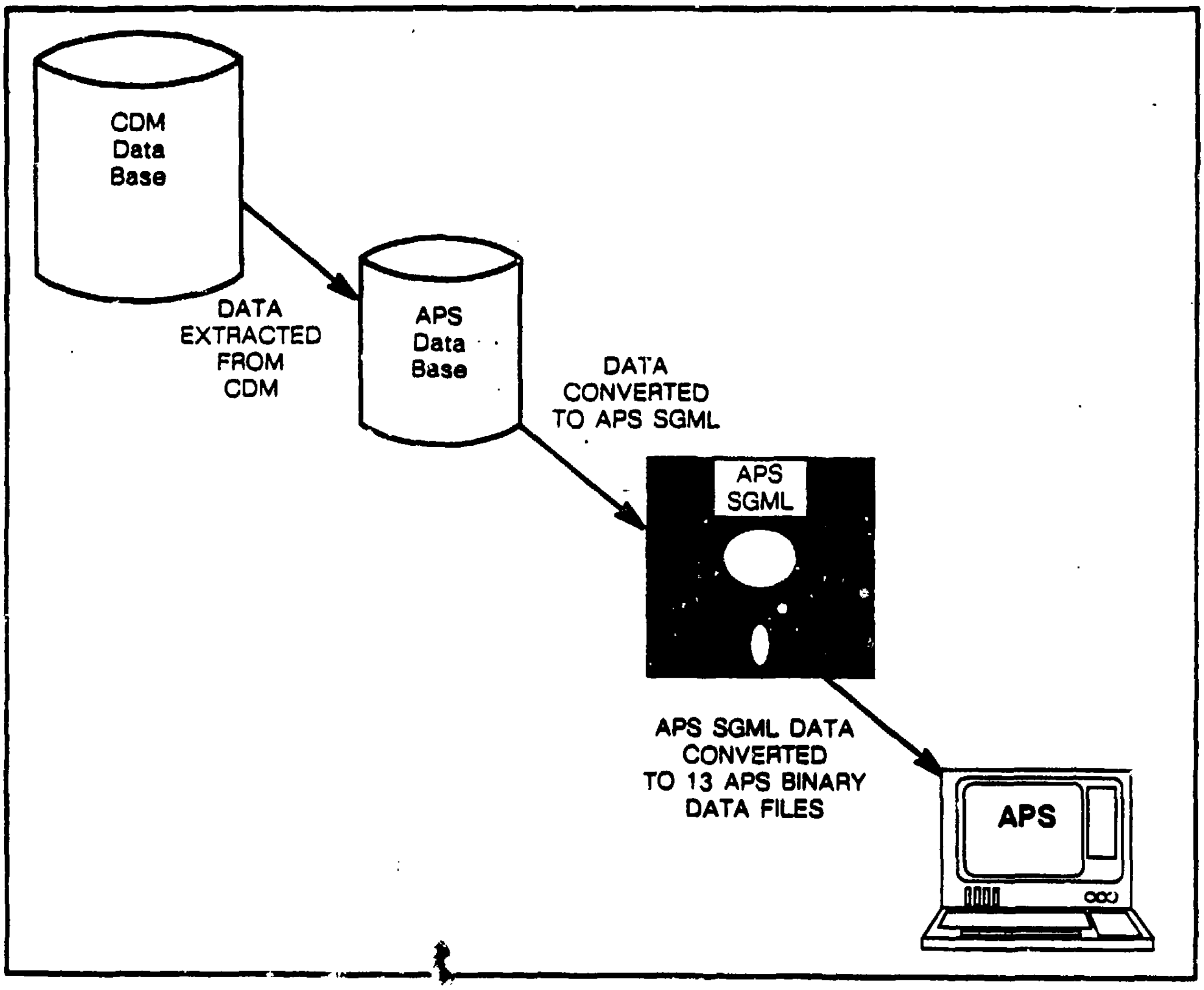


Figure 21. CDM to APS Conversion.

Results of Limited Demonstration/Validation

AFHRL, RJO, and Datalogics conducted a formal demonstration/tutorial in December, 1988. In this demonstration, Air Force and contractor personnel were briefed on the work that had been accomplished to that point. Demonstrations also provided evidence that data could be loaded into and extracted from the CDM in many different formats. Conversions from APS and ITDS also showed that data developed and stored using different systems and different data base designs could be compatible with the CDM. Future plans were also discussed. These plans included the further testing of the CDM and the development of documentation to accompany the CDM specification. Figure 22 illustrates the entire process used in the limited demonstration and validation of the CDM.

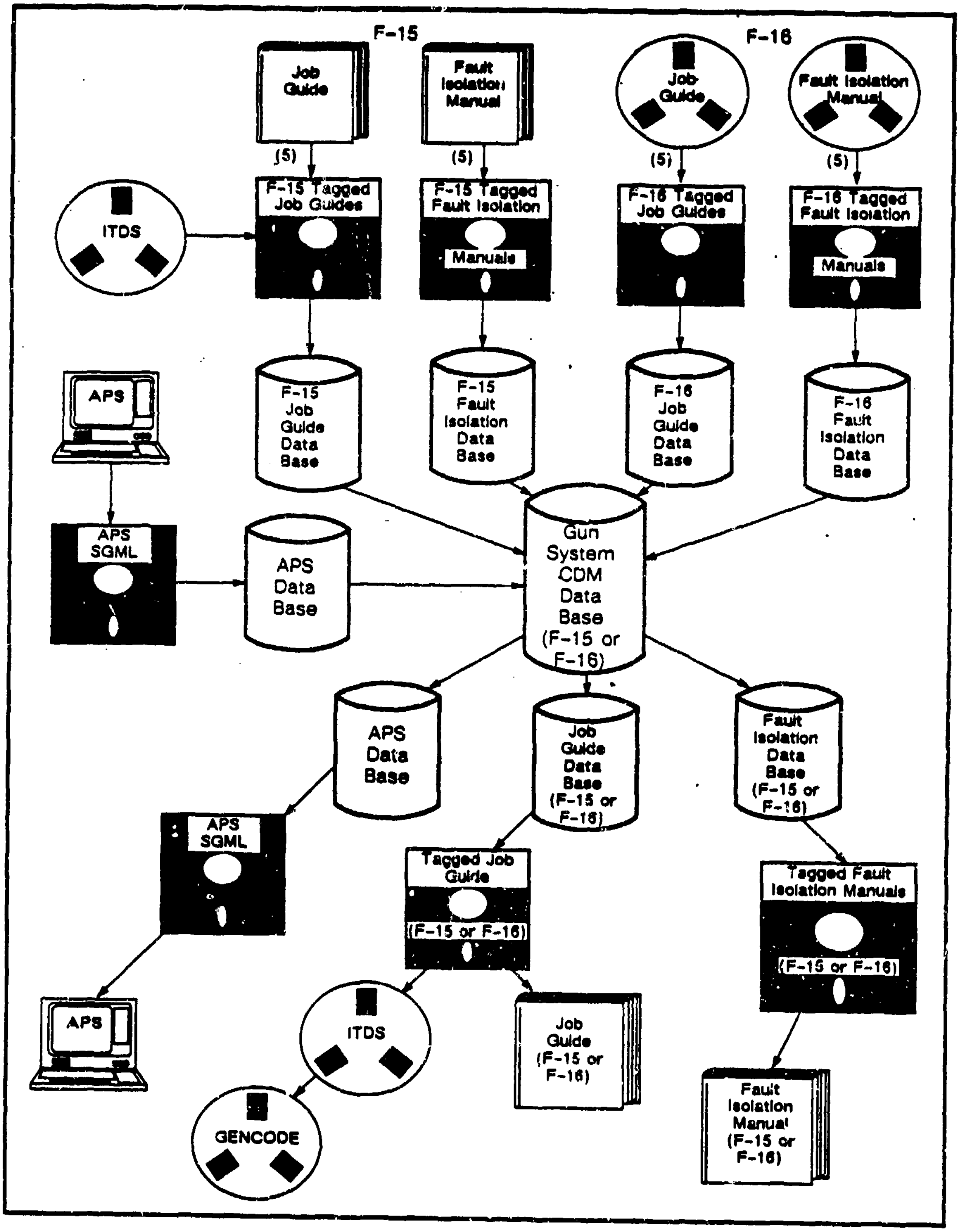

F-15
F-16
Job Guide
Fault Isolation Manual
Job Guide
Fault Isolation Manual
(5)
(5)
(5)
(5)
ITDS
F-15 Tagged Job Guides
F-15 Tagged Fault Isolation Manuals
F-16 Tagged Job Guides
F-16 Tagged Fault Isolation Manuals
APS
F-15 Job Guide Data Base
F-15 Fault Isolation Data Base
F-16 Job Guide Data Base
F-16 Fault Isolation Data Base
APS SGML
APS Data Base
Gun System CDM Data Base (F-15 or F-16)
APS Data Base
Job Guide Data Base (F-15 or F-16)
Fault Isolation Data Base (F-15 or F-16)
APS SGML
Tagged Job Guide (F-15 or F-16)
Tagged Fault Isolation Manuals (F-15 or F-16)
APS
ITDS
Job Guide (F-15 or F-16)
GENCODE
Fault Isolation Manual (F-15 or F-16)

Figure 22. Limited Demonstration/Validation.

Full-Scale Demonstration/Validation

Following the limited demonstration and validation, the CDM was redesigned to accommodate the data and structures found in intermediate and depot level manuals (see Section III).

Data Selection

To further validate the CDM as a means of maintaining technical data for an entire weapons system, it was necessary to load all of the data for a specific subsystem within a weapon system into the CDM. This further validation involved the inclusion of the intermediate level and depot level maintenance information as well as the remainder of the organizational level information. In this phase, information was used for the M61A1 and M61 automatic gun system on the F-15 only. It did not seem beneficial to also perform further validation on the F-16 data since the previous variations seemed to be primarily format oriented.

Information included in the data base was entered from the following types of manuals:

- Organizational

  Job Guide (42 sections)
  Fault Isolation Manual (26 fault codes)
  General System Manual
  General Vehicle Manual
  Checklist

- Intermediate

  Intermediate Maintenance Instructions
  Illustrated Parts Breakdown

- Depot

  Overhaul Instructions

Data Mapping

A revised DTD was developed for the CDM which described the structure of the technical information more accurately to accommodate differences in intermediate and depot level data. The job guide and fault isolation DTDs were also modified slightly. DTDs were developed for intermediate maintenance instructions and overhaul instruction manuals as well.

Job Guide DTD. Differences in structure and format were found in the additional job guides which required enhancements to the original job guide DTD. The DTD was revised to model these differences.

Fault Isolation Manual DTD. The fault isolation manual DTD was also enhanced as needed for the additional fault isolation manuals.

Intermediate Maintenance Instructions DTD. The DTD for intermediate level maintenance instructions was developed using the MIL-M-38784 compliant DTD contained in MIL-M-28001. MIL-M-6675 was used to develop the structures while examination of existing intermediate maintenance manuals provided other structural information. Since manuals from only one weapon system were being examined, no format differences were discovered from manual to manual. Appendix D contains the intermediate maintenance instructions DTD.

Overhaul Instructions DTD. The development of the overhaul instruction DTD used the MIL-M-38784 compliant DTD contained in MIL-M-28001 as a baseline. MIL-M-38789 was used to develop structures as well as examine existing overhaul manuals. Appendix E contains the overhaul instruction DTD.

Data Entry

Data used for the specification development phase of the effort was entered using a word processor on an IBM PC. Graphics were also received from McDonnell Douglas Aircraft Company in the Initial Graphics Exchange Standard (IGES) format. IGES is a standard for the interchange of vector graphics. These graphics were the most current graphics contained in the manuals required for maintenance on the gun system. In some cases, the graphics did not coincide with the source information supplied for the development of the CDM. Figure 23 illustrates the process used to enter data.

Direct Entry. The textual data for the five job guide and five fault isolation manuals developed during the demonstration and validation phase was enhanced. The enhancements included the revision of updated tags and the definition of previously undefined attributes. The additional job guides and fault isolation manuals as well as the data contained in the intermediate maintenance instructions and overhaul manuals was entered using a word processing program on an IBM PC.

Tagging. The data entered in the previous step was tagged in accordance with the appropriate DTD using an ASCII text editor. Tags for the inclusion of the IGES graphics were also added to the tagged document instances.

Parsing. As was done in the demonstration and validation phase, a parser checked each file to verify conformance to the DTD and to SGML.

Data Loading

The revised CDM structure required that the loaders developed in the Demonstration and Validation phase be revised. Also, the development of an automatic combining program facilitated the development of small CDM files which could be merged into one large CDM data file.

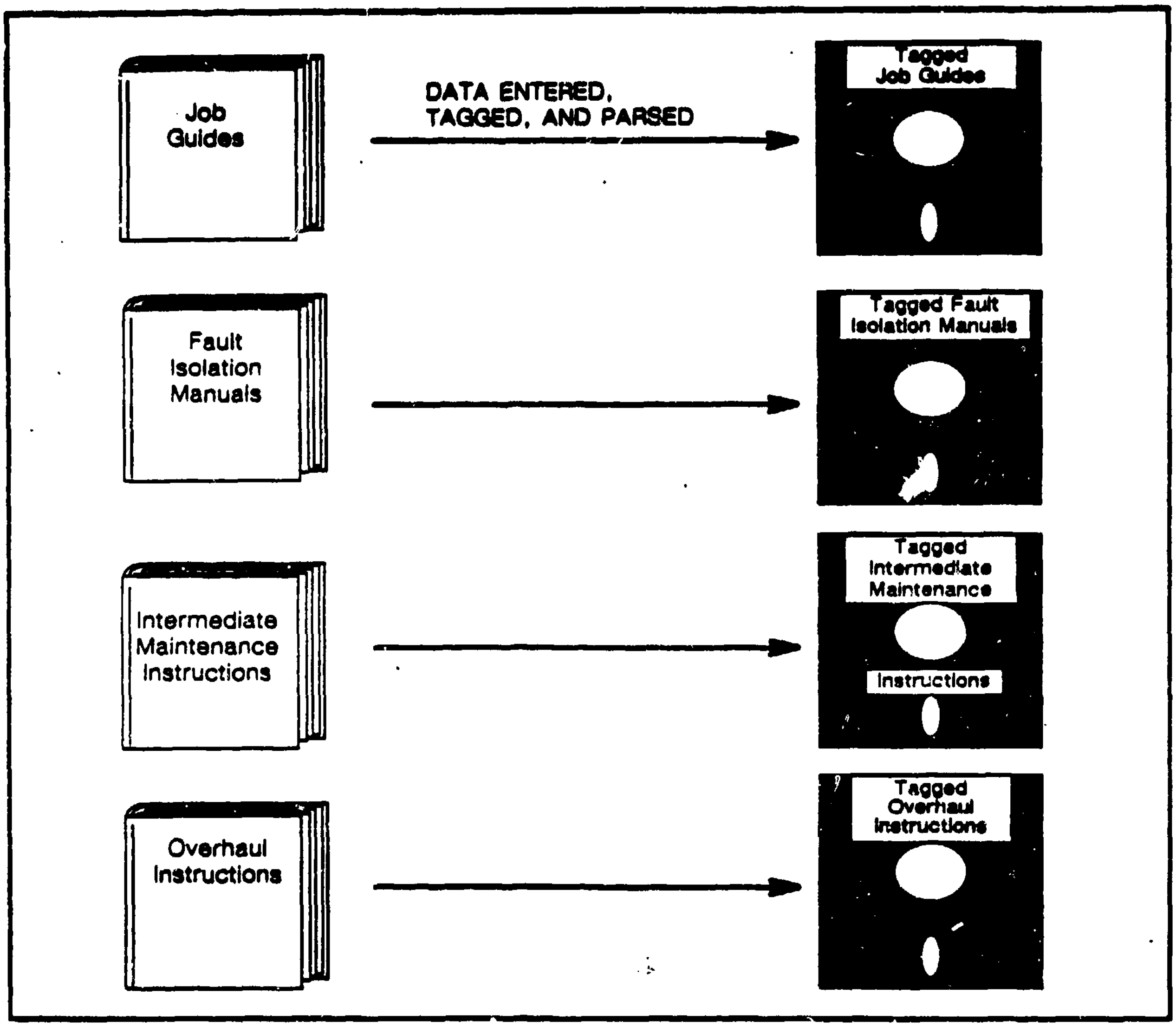

Job Guides
DATA ENTERED, TAGGED, AND PARSED
Tagged Job Guides
Fault Isolation Manuals
Tagged Fault Isolation Manuals
Intermediate Maintenance Instructions
Tagged Intermediate Maintenance
Instructions
Overhaul Instructions
Tagged Overhaul Instructions

Figure 23. Data Entry for Full Scale Demonstration/Validation.

Direct Loading. Data found in manuals for which DTDs were not developed (i.e., Checklist, General System, General Vehicle, Illustrated Parts Breakdown) was tagged directly into the CDM structure (see Figure 24). The volume of data specific to the gun system did not warrant the development of a separated DTD for each manual and it was determined that data could be tagged directly into the CDM structure. This was done using an ASCII text editor. The data was successfully entered into the structure but the robustness of the attributes (i.e., cross references, etc.) was not equal to the data which had been loaded from the other DTDs. Tags for the inclusion of the IGES graphics were also included in the manually developed CDM text file.

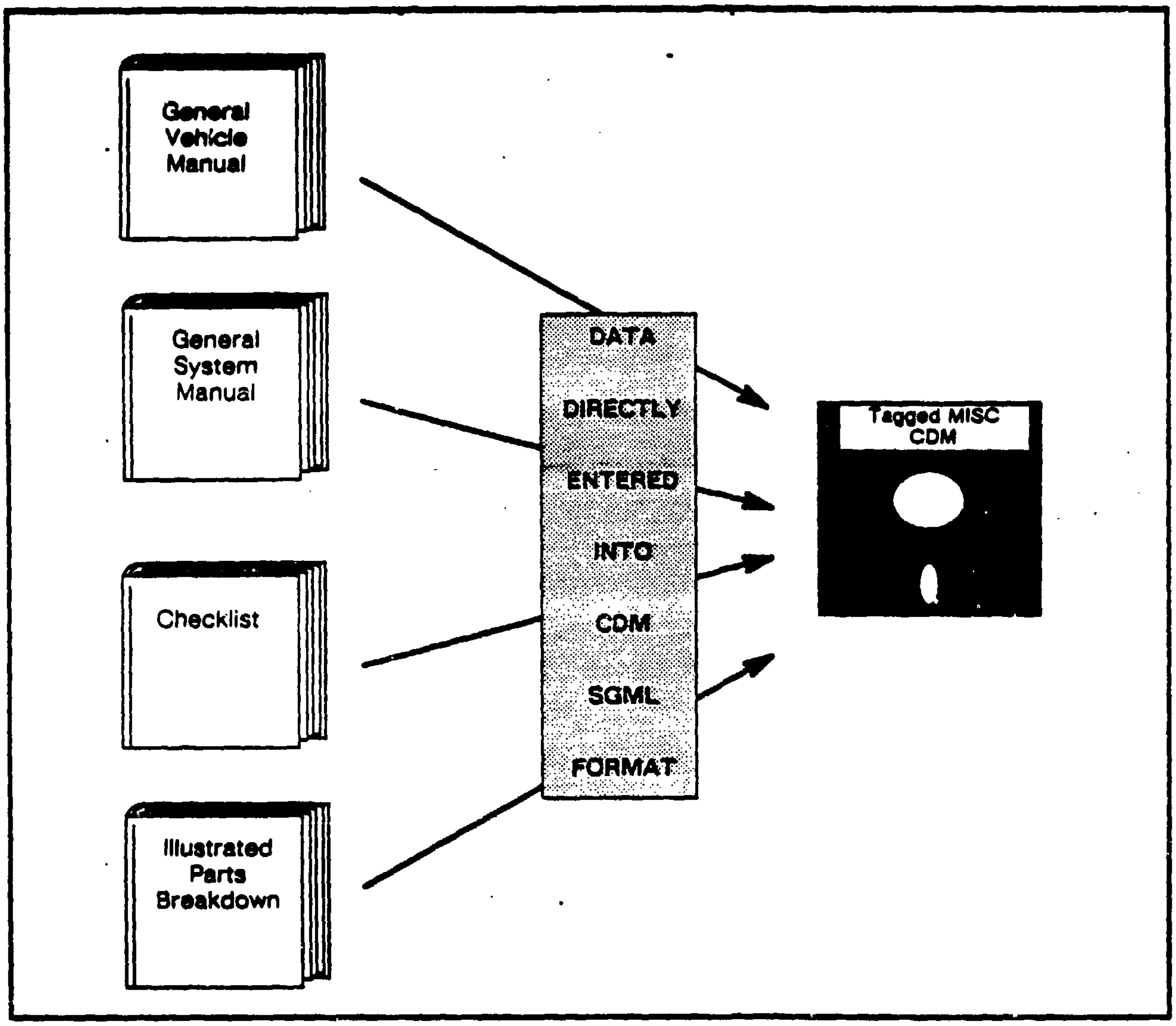

General Vehicle Manual
General System Manual
Checklist
Illustrated Parts Breakdown
DATA DIRECTLY ENTERED INTO CDM SGML FORMAT
Tagged MISC CDM

Figure 24. Direct Loading Into the CDM.

Loading. The files containing the job guide, fault isolation, intermediate maintenance, and overhaul data were loaded into a program developed using the General Expression Pattern Recognition (GEPR) language. This program converted the text files into a CDM text file. The information contained in each type of manual was loaded into a separate CDM text file. The CDM text files, including the manually developed file, were then converted into CANDOR data base tables for processing as had been done in the demonstration and validation phase. Figure 25 shows the processes used for loading data into the CDM structure.

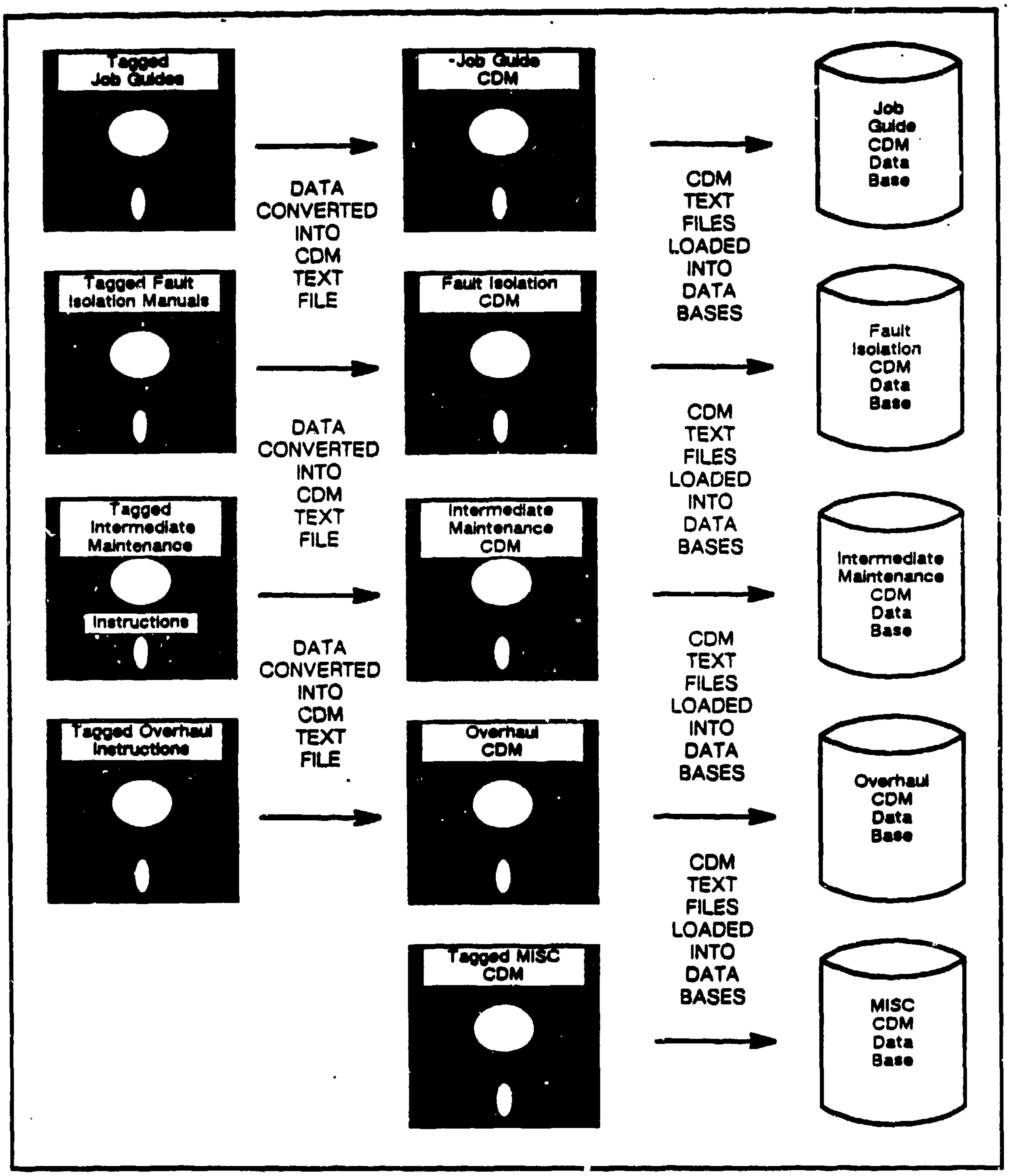

Tagged Job Guides
DATA CONVERTED INTO CDM TEXT FILE
-Job Guide CDM
CDM TEXT FILES LOADED INTO DATA BASES
Job Guide CDM Data Base
Tagged Fault Isolation Manuals
DATA CONVERTED INTO CDM TEXT FILE
Fault Isolation CDM
CDM TEXT FILES LOADED INTO DATA BASES
Fault Isolation CDM Data Base
Tagged Intermediate Maintenance Instructions
DATA CONVERTED INTO CDM TEXT FILE
Intermediate Maintenance CDM
CDM TEXT FILES LOADED INTO DATA BASES
Intermediate Maintenance CDM Data Base
Tagged Overhaul Instructions
Overhaul CDM
CDM TEXT FILES LOADED INTO DATA BASES
Overhaul CDM Data Base
Tagged MISC CDM
MISC CDM Data Base

Figure 25. Loading of Data Into CDM Data Bases.

Combining. In order to get a complete CDM containing the information from all of the manuals, the individual CDM CANDOR tables for each manual were combined (see Figure 26). This was done by a program developed using GEPR which designated one of the CANDOR tables as the "master." The program was run on this master table which eliminated any redundancy within it. The same program was then run on the other CANDOR tables to be combined with the "master" table to create a complete CDM CANDOR instance.

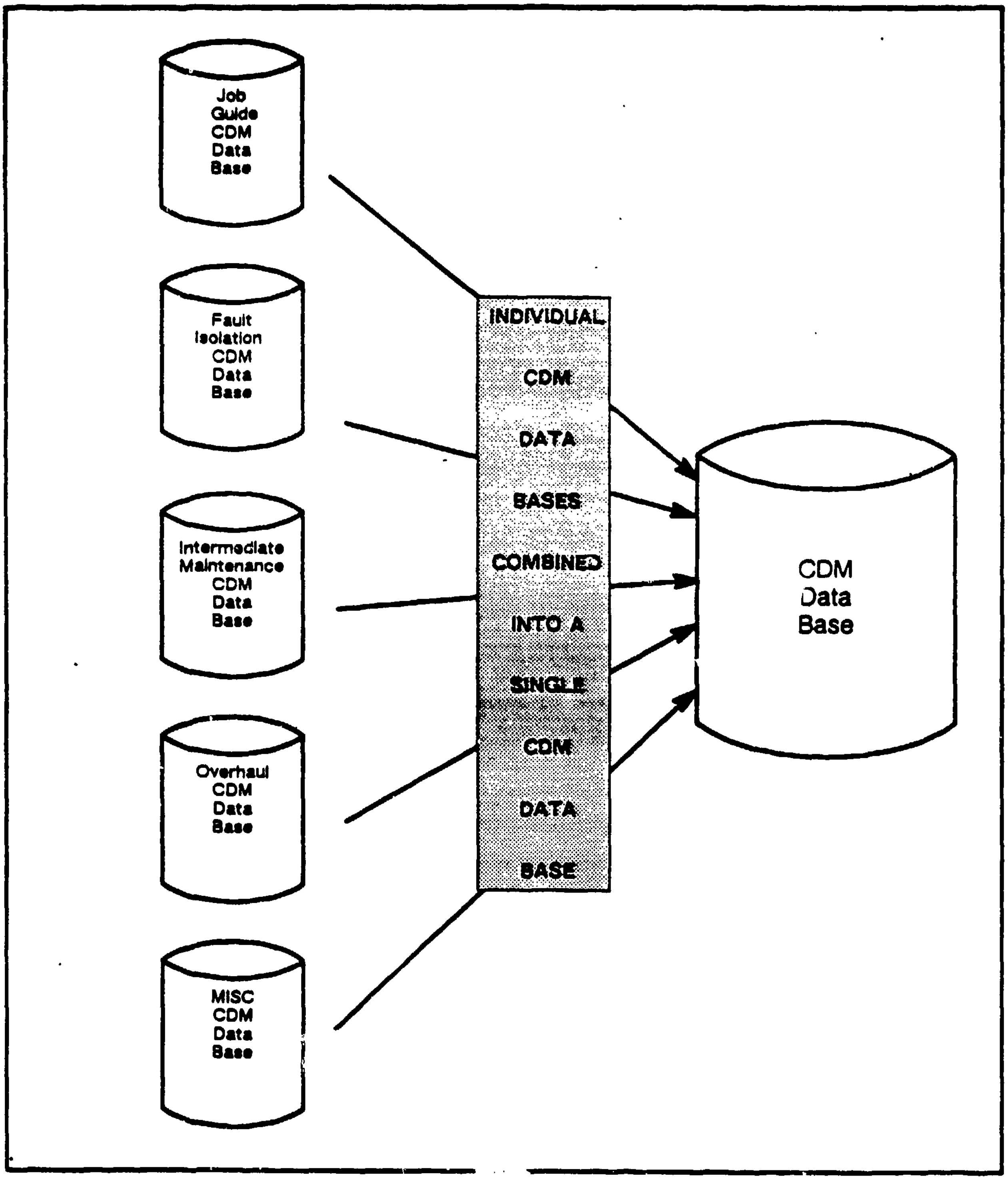

Job Guide CDM Data Base
Fault Isolation CDM Data Base
Intermediate Maintenance CDM Data Base
Overhaul CDM Data Base
MISC CDM Data Base
INDIVIDUAL CDM DATA BASES COMBINED INTO A SINGLE CDM DATA BASE
CDM Data Base

Figure 26. Combining of CDM Data Bases Into Single CDM.

Data Extraction

· The data extraction process used in the full scale development and validation phase was similar to that used in the limited demonstration and validation phase. Based on the type of output desired, a GEPR program extracted the required information from the CANDOR tables using the DTD as a guide. The output consisted of an SGML-tagged text file which corresponded with the appropriate DTD. Figure 27 illustrates the data extraction process.

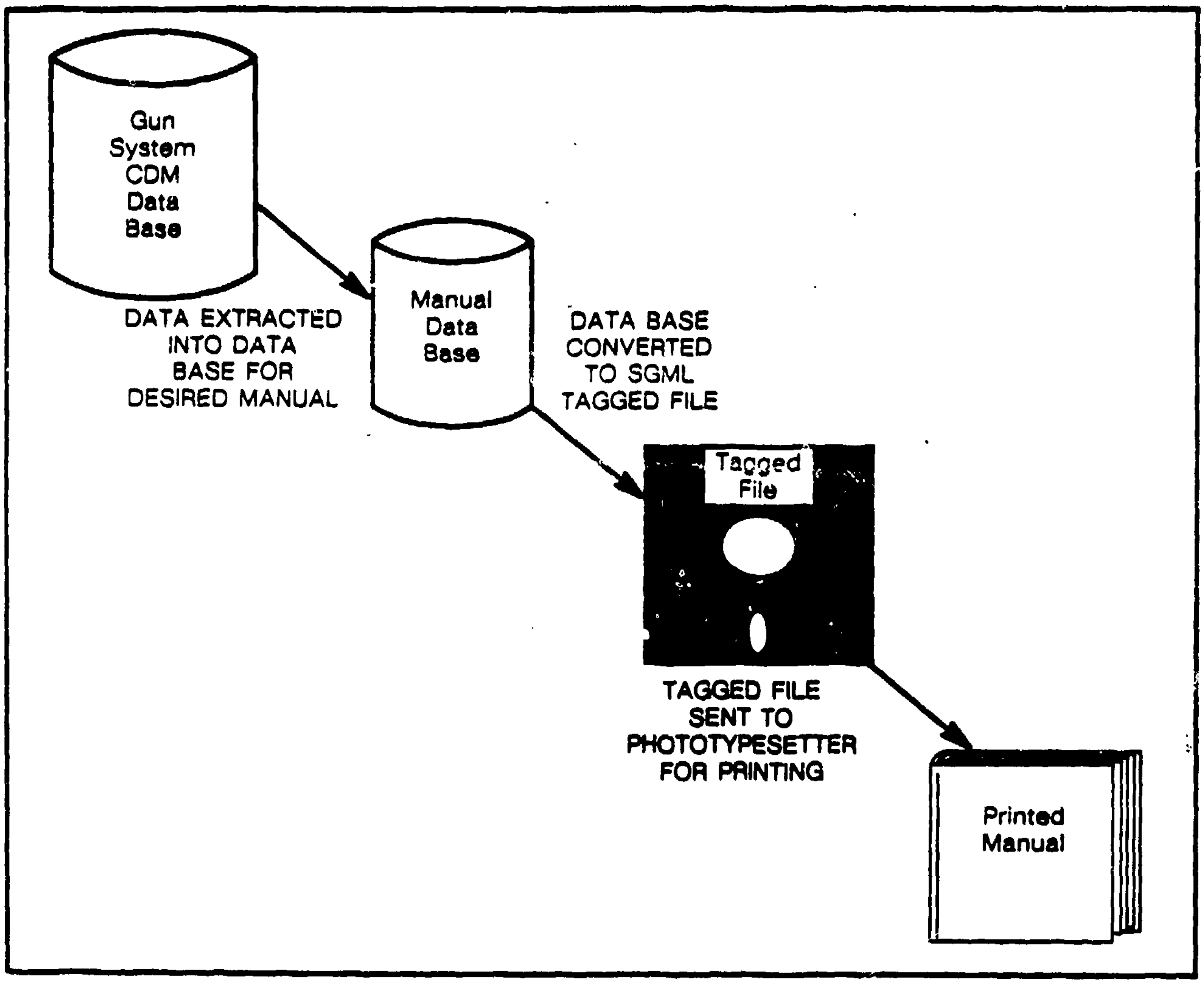

Gun System CDM Data Base
DATA EXTRACTED INTO DATA BASE FOR DESIRED MANUAL
Manual Data Base
DATA BASE CONVERTED TO SGML TAGGED FILE
Tagged File
TAGGED FILE SENT TO PHOTOTYPESETTER FOR PRINTING
Printed Manual

Figure 27. Data Extraction For Full Scale Demonstration/Validation.

In order to perform quality checks on the extracted information, a phototypesetter was again used.

Results of Full-Scale Demonstration/Validation

AFHRL, RJO, and Datalogics conducted a second formal demonstration/tutorial in August 1989. In this demonstration, Air Force and contractor personnel were updated on the work accomplished to that point. A great deal of emphasis was placed on the development of the implementation specification. Plans were made for the review of the specification by government and by industry. In addition, a detailed tutorial on the CDM was presented in order to assist those in attendance in the development of CDM data bases. Figure 28 shows the processes used during the full scale development and validation of the CDM.

Summary of Results

The CDM validation efforts showed that a data base for technical information could be developed. They also showed that the data base could be loaded in a number of ways and data could also be extracted in a number of ways. Data developed and stored using other systems could also be loaded into the CDM as well as from the CDM.

V. DOCUMENTATION DEVELOPMENT

During this effort, several interim documents were produced which documented the progress of the research. These are listed in the references section. Two of the most important documents produced during this effort include the CDM Exchange Standard and the CDM Implementation Specification.

CDM Exchange Standard

Objective

A data exchange standard was developed which defined the structure of the CDM data base as it will be exchanged under MIL-STD-1840. This standard was based on the CDM DTD. It dictates the interchange of CDM data similar to the way in which MIL-M-28001 dictates the format in which the current paper manuals are interchanged.

Scope

At this time, the CDM Exchange Standard addresses the interchange of maintenance and operational information. Training, manufacturing and logistics information are not included in the CDM at this time but are expected to be added in the future.

Contents

The exchange standard contains the CDM DTD which will be used for the interchange of technical information under MIL-STD-1840. An appendix is also included which provides descriptions of all the data element tags and attributes. This was included to aid the user who is unfamiliar with SGML and the CDM.

Specification for Digital Technical Information

A draft specification for digital technical information was developed which outlined the creation, distribution, processing, and use of digital technical information designed to support the traditional functions of operation and maintenance.

Objective

The specification establishes the requirements for the creation, delivery, and presentation of integrated technical information. It includes requirements for:

1. general information content, authoring style, display presentation, and user interaction.

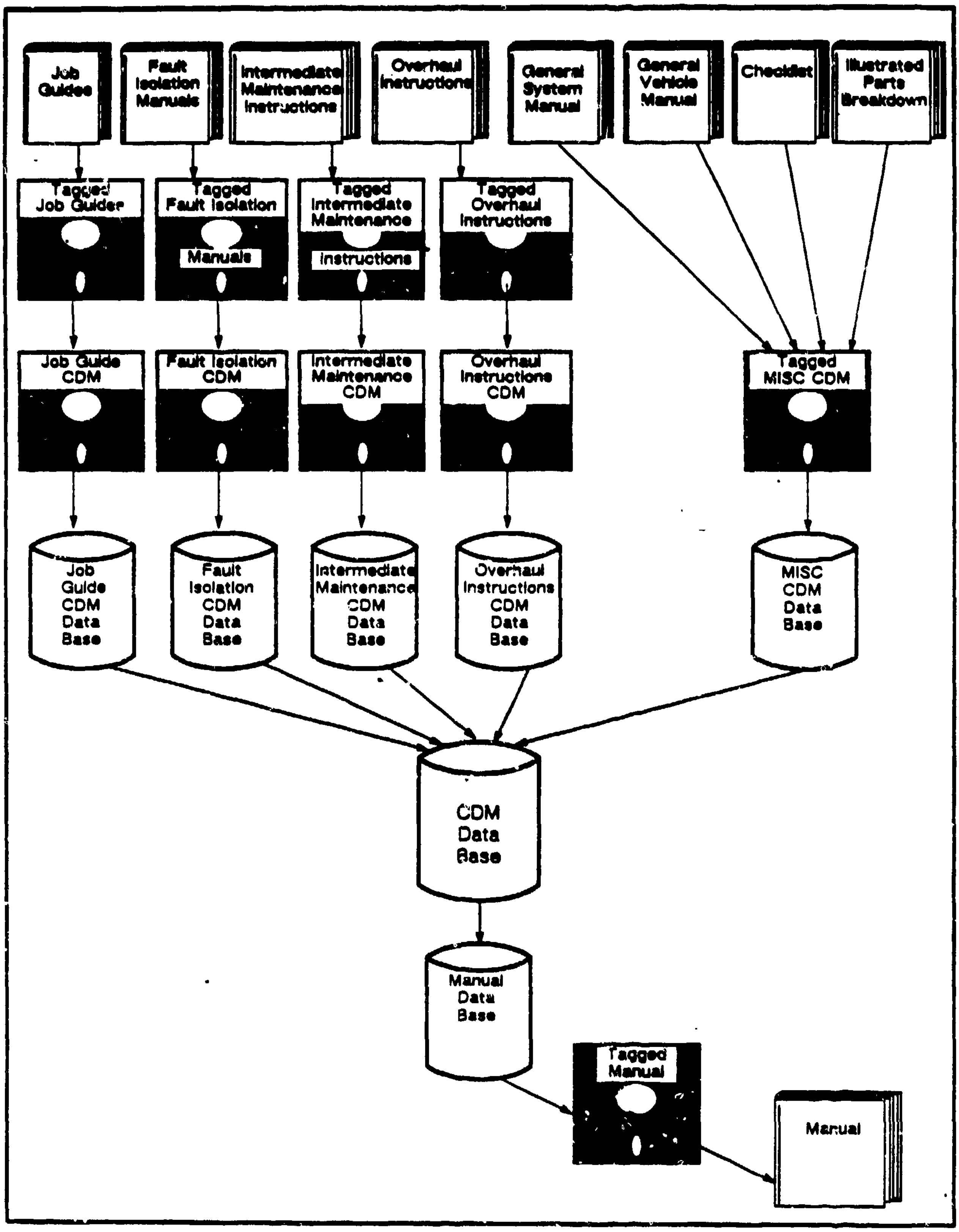

Job Guides
Fault Isolation Manuals
Intermediate Maintenance Instructions
Overhaul Instructions
General System Manual
General Vehicle Manual
Checklist
Illustrated Parts Breakdown
Tagged Job Guides
Tagged Fault Isolation Manuals
Tagged Intermediate Maintenance Instructions
Tagged Overhaul Instructions
Job Guide CDM
Fault Isolation CDM
Intermediate Maintenance CDM
Overhaul Instructions CDM
Tagged MISC CDM
Job Guide CDM Data Base
Fault Isolation CDM Data Base
Intermediate Maintenance CDM Data Base
Overhaul Instructions CDM Data Base
MISC CDM Data Base
CDM Data Base
Manual Data Base
Tagged Manual
Manual

Figure 28. Full Scale Demonstrati‿n/Validation.

2. **revisable, format-free technical information.**

3. **view packages of technical information.**

4. **functional presentation system hardware and software requirements.**

5. **quality assurance.**

6. **interchange.**

Scope

The specification outlines how to create, distribute, process, and use integrated technical information to support the traditional functions of operation and maintenance.

Contents

The specification is divided into six sections which establish the requirements an acquisition activity shall consider when procuring integrated technical information. These requirements cover the entire life-cycle of the information from creation, to presentation, to delivery and, finally, validation.

General Information Content, Authoring Style, Display Presentation, and User Interaction Requirements. Since digital technical information can be prepared in a number of ways and can also be used in many different ways, minimum standards for the authoring, display, and use of technical information are specified. These standards include the informational content, authoring style guidance, presentation, and user interface requirements for digital technical information.

Revisable Technical Information. To facilitate the maintenance and operation of equipment, technical information may be presented on a variety of media, formatted to comply with a variety of specifications. Even though the information may be used in so many ways, the underlying data elements required to support maintenance and operation of a particular piece of equipment is basically the same for all equipment.

View Packages of Digital Technical Information. A view package is a subset of the digital technical information that supports a specific objective. The information is collected and formatted based on the requirements of a specific function. The actual information and the formatting rules make up the view package. It should also provide basic guidelines for use in developing specific application requirements.

Presentation System Requirements. A presentation system will be used to display view packages of technical information to the users of the information. Types of presentation systems will vary greatly. Possible presentation systems include: phototypesetters, screen frame paging systems, interactive color workstations, and hand-held miniature computers. No matter what type of presentation system is utilized, they all have a common purpose: to present the technical information contained in a view package, using the presentation rules associated with that view package, so that the final user of the information can retrieve accurate, understandable technical information.

Quality Assurance Requirements. Quality assurance is performed on the technical information in two ways. First, the revisable technical information data base may be checked for several items including: readability; accuracy of attributes; appropriateness of cross-references; naming schemes; flow of the information; appropriateness and correctness of warnings, notes, and cautions; and appropriateness of context attributes.

Second, the output of the technical information will be checked for the content of the data, the arrangement of the information on the display medium, and understandability of the presented information.

Interchange Requirements. Digitally represented technical information can be interchanged efficiently between dissimilar computer systems using MIL-STD-1840 which provides a framework for the interchange of digital technical information as well as addressing media types and packaging of the technical information. Additions will need to be made to MIL-STD-1840 to govern the exchange of the revisable technical information, view packages, and presentation system software.

VI. SUMMARY/RECOMMENDATIONS

These Research and Development (R&D) efforts have shown the CDM will work. With today's technologies, a single data base can be developed that supports O-Level, I-Level and D-Level maintenance. This single data base can store all of the required content and structure with minimal or no data redundancy. The format of the maintenance information can be stored outside of the data base and will be different for each output device. A single definition of format can be established for each output device and can be used for any system's maintenance data base.

The potential savings associated with this concept are staggering. With this type of technology, the maintenance data base will not have to be changed when different output devices are developed. With this model, the USAF could revolutionize technical information far beyond current programs in development.

However, there are still significant amounts of R&D which should to be performed. The benefits of the CDM have to be verified. This verification would include evaluating the system when it is being used in the field as well as doing Life Cycle Cost estimates to show the expected savings.

Finally, the CDM must be continually studied and reviewed. Such review would allow the CDM to adapt to a rapidly changing environment. A two-phased approach for review may be used. The first phase would focus on continued population of the CDM with even more data to further assure the ability to implement the design. The second phase would involve periodic reviews by the community. Without these continual reviews by the DoD and major contractors, the CDM will not be proposed or implemented correctly.

This paper has described the evolution of the CDM. Detailed descriptions and explanations are given to assist the reader in understanding the concepts involved in the CDM. This paper has also described efforts to demonstrate and validate the CDM concept and develop specifications for the exchange of the technical information.

REFERENCES

Bryan, Martin. (1988). SGML - An author's guide to the standard generalized markup language. Reading, MA: Addison-Wesley Publishing Company.

Datalogics, Inc. (1988, July). Final Report - Content Data Model of organizational maintenance information for automated interchange of technical source data. Unpublished manuscript, Wright-Patterson AFB, OH: Logistics and Human Factors Division, Air Force Human Resources Laboratory.

Earl, Mark J. (1989, January). The Content Data Model - A specification for integrated data bases of technical information. Unpublished manuscript, Wright-Patterson AFB, OH: Logistics and Human Factors Division, Air Force Human Resources Laboratory.

ISO-8879-1986. (1986). Information processing - Text and office systems - Standard Generalized Markup Language. International Standards Organization.

Link, William R., Von Holle, Joseph C., & Mason, Dwayne. (1987, November). Integrated Maintenance Information System (IMIS): A maintenance information delivery concept (AFHRL-TP-87-27, AO-A189 335). Wright-Patterson AFB, OH: Logistics and Human Factors Division, Air Force Human Resources Laboratory.

MIL-D-28003. (1988). Digital representation for communication of illustration data: CGM application profile. Washington, DC: United States Department of Defense.

MIL-HDBK-59. (1988). Computer-Aided Acquisition and Logistics Support (CALS) program implementation guide. Washington, DC: United States Department of Defense.

MIL-M-28001. (1988). Markup requirements and generic style specification for electronic printed output and exchange of text. Washington, DC: United States Department of Defense.

MIL-M-38784. (1983). Manuals, technical: General style and format requirements. Washington, DC: United States Air Force.

MIL-M-38789. (1973). Manuals, technical: Overhaul instructions and overhaul instructions with illustrated parts breakdown (for various equipment and accessories). Washington, DC: United States Air Force.

MIL-M-6675. Manual, technical: Intermediate maintenance instructions. Washington, DC: United States Air Force.

MIL-M-83495. (1977). Manuals, technical, Organizational maintenance manual set: General requirements for preparation of (for aircraft, missiles, and space vehicles). Washington, DC: United States Air Force.

MIL-R-28002. (1988). Raster graphics representation in binary format, requirements for. Washington, DC: United States Department of Defense.

MIL-STD-1840. (1987). Automated interchange of technical information. Washington, DC: United States Department of Defense.

RJO Enterprises, Inc. (1988, November). Improved Technical Data System (ITDS) to Content Data Model (CDM) data conversion routine. Computer software product. Wright-Patterson AFB, OH: Logistics and Human Factors Division, Air Force Human Resources Laboratory.

RJO Enterprises, Inc. (1988, November). Interim report on R & D - Entering test data. Unpublished manuscript. Wright-Patterson AFB, OH: Logistics and Human Factors Division, Air Force Human Resources Laboratory.

RJO Enterprises, Inc. (1988, December). Interim report on R & D - Extracting test data. Unpublished manuscript. Wright-Patterson AFB, OH: Logistics and Human Factors Division, Air Force Human Resources Laboratory.

RJO Enterprises, Inc. (1989, January). CDM - Data conversion methods - APS data. Unpublished manuscript. Wright-Patterson AFB, OH: Logistics and Human Factors Division, Air Force Human Resources Laboratory.

RJO Enterprises, Inc. (1989, January). Content Data Model (CDM) Job Guide DTD - ITDS - Gencode data conversion process. Unpublished manuscript. Wright-Patterson AFB, OH: Logistics and Human Factors Division, Air Force Human Resources Laboratory.

RJO Enterprises, Inc. (1989, August). Draft specification for digital technical information. Unpublished manuscript. Wright-Patterson AFB, OH: Logistics and Human Factors Division, Air Force Human Resources Laboratory.

RJO Enterprises, Inc. and Datalogics, Inc. (1988, October). Document Type Definition for fault isolation manuals as defined in MIL-M-83495. Unpublished manuscript. Wright-Patterson AFB, OH: Logistics and Human Factors Division, Air Force Human Resources Laboratory.

RJO Enterprises, Inc. and Datalogics, Inc. (1988, May). Document Type Definition for Jobguide manuals as defined in MIL-M-83495. Unpublished manuscript. Wright-Patterson AFB, OH: Logistics and Human Factors Division, Air Force Human Resources Laboratory.

Shroder, Ron, Teall, Thomas A., and Millhouse, Robert. (1989, February). Content Data Model (CDM) and Improved Technical Data System (ITDS) compatibility analysis AFHRL-TP-88-43, AD-A205 916). Wright-Patterson AFB, OH: Logistics and Human Factors Division, Air Force Human Resources Laboratory.

Thomas, Donald L. and Clay, Jeffery D. (1988, August). Computer-Based maintenance aids for technicians: Project final report (AFHRL-TR-87-44, AD-A198 283). Wright-Patterson AFB, OH: Logistics and Human Factors Division, Air Force Human Resources Laboratory.

GLOSSARY

The following are acronyms and abbreviations used throughout this report:

AFHRL	Air Force Human Resources Laboratory
AFLC	Air Force Logistics Command
AFSC	Air Force Systems Command
AIA	Aerospace Industries Association
APS	Authoring and Presentation System
ASCII	American Standard Code for Information Interchange
ATOS	Automated Technical Order System
CAD	Computer Aided Design
CALS	Computer-Aided Acquisition and Logistics Support
CANDOR	Canonical Document Representation
CDM	Content Data Model
CMAS	Computer-Based Maintenance Aiding System
DoD	Department of Defense
DTD	Document Type Definition
DXF	Drawing Interchange File
GEPR	General Expression Pattern Recognition
IGES	Initial Graphics Exchange Standard
IMIS	Integrated Maintenance Information System
IPB	Illustrated Parts Breakdown
ISO	International Standards Organization
ITDS	Improved Technical Data System
NIST	National Institute of Standards and Technology (formerly National Bureau of Standards)
NTIPS	Naval Technical Information Publishing System
SGML	Standard Generalized Markup Language
SSSN	System, Subsystem, Subject Number

APPENDIX A

CONTENT DATA MODEL
DOCUMENT TYPE DEFINITION

```
<!DOCTYPE  techinfo[

<!--
*****************************************************************

          Content Data Model (CDM)
               Version 5.1  8/23/89  DRAFT

This document is an SGML Document Type Definition (DTD) which
describes the logical structure for a database of technical
information.  This version is a working draft which is still under
development by the Air Force Human Resources Laboratory (AFHRL/LRC)
at Wright Patterson, AFB.

Changes from Version 5.01 are indicated in bold face type

**************************************************************-->
```

```
<!- *******************************************************************
```

Entity Declarations

These entity declarations define abbreviations for a set of frequently used attributes: "id", "refid".

The "%ids" entity declaration defines an abbreviation for two element identifiers ("id" and "refid"). These identifiers are machine generated symbols used to identify data elements in the database. These identifiers are not intended to be human readable names, but only machine readable pointers or references. Most elements in the CDM have both a unique identifier ("id") and a non-unique reference identifier ("refid"). The reference identifier ("refid") is used by elements to refer or "point" to other elements. For example, a "task" element may refer to a list of "step" elements, or a "system" element may refer to a list of "subassembly" elements. Since the reference id ("refid") is not unique, it may refer to several elements in the database. These referenced elements will have different "context" (such as version number or security level) which will be used to determine which unique element is appropriate for a particular situation. Since SGML requires that all "ID" attributes be unique, an "ID" attribute cannot be used as the reference id ("refid"). Instead, "refid" is defined as a NMTOKEN.

The two entities "%refid" and "%refids" are defined only to improve readability. They let the reader know that an attribute's value is meant to be a reference identifier (i.e., a NMTOKEN specifying some element's "refid"). These two entities, "%refid" and "%refids", are used throughout the CDM whenever a reference is intended. This DTD also follows the convention that the attribute name of any "%refid" attribute is the same as the element name to which it is referring. For example, the "system" element has an attribute named "task" which is a "%refids" list intended to contain only valid "refids" to "task" elements. There is on notable exception to this rule. In some cases an "elmntref" attribute is defined which is intended to be a reference to any element type.

```
*********************************************************************->

<!ENTITY  % ids     "id  ID     #REQUIRED
          refid     NMTOKEN    #REQUIRED">

<!ENTITY  % refid   "NMTOKEN">

<!ENTITY  % refids  "NMTOKENS">
```

```
<!--*******************************************************************

          Notation Declarations

The following notations define external references to "public"
graphics standards used in the CDM.  The specified abbreviations
(cgmbin, cgmclear, cgmchar, fax, iges, dxf, gks) are used by the
element "graphprm" to specify the type of graphic representation
used to encode a particular graphic primitive.

**********************************************************************
******************************************-->

<!NOTATION   cgmbin PUBLIC   "ISO   8632/2//NOTATION   Binary
encoding//EN">

<!NOTATION   cgmchar    PUBLIC "ISO 8632/2//NOTATION Character
encoding//EN">

<!NOTATION   cgmclear   PUBLIC "ISO 8632/2//NOTATION Clear text
encoding//EN">

<!NOTATION   fax    PUBLIC "-//USA-DOD//NOTATION CCITT Group 4
Facsimile//EN">

<!NOTATION   iges   PUBLIC "-//USA-DOD//NOTATION Initial Graphics
Exchange Specification//EN">

<!NOTATION   dxf    PUBLIC "-//USA-DOD//NOTATION DXF Encoding//EN"
>

<!NOTATION   gks    PUBLIC "-//USA-DOD//NOTATION Graphics Kernel
System//EN" >
```

```
<!--**************************************************************

                         CDM Content
"Techinfo"  is the top element of the CDM .   Its content model is
a long list of elements (vehicle*, system*, ... ) which comprise
the raw data "records" or "tables" for  the CDM data base.  The
first "system" element in the data base is the root or top level
item in the hierarchy.

***************************************************************-->


<!ELEMENT techinfo  - -  ( system*, operinfo*, descinfo*,
               task*, input*,  step*,
               reqcond*, person*, equip*, consum*, verb*,
               warning*, caution*, note*,  annot*,
               partinfo*, partbase*, faultinf* ,
               test*, outcome*,  fltstate*, fault*, rect*,
               text*, title*, dictitem*, xref*, table*, list*,
               graphic*,  tgraphic*, grphprim*,
               video*, audio*, process*,
               prompt*, fillin*, menu*, choice*,
               assertion*, precond*, property*, value* )>
```

```
<!--*****************************************************************
*
```

Vehicle - System - Subsystem Hierarchy

The CDM specifies a hierarchically organized data base of technical information for a weapon system. The main hierarchy of the data base parallels the equipment hierarchy of the weapon system. This is no mally represented as a vehicle - system - subsystem - subassembly hierarchy in AF Technical Orders. That hierarchy is represented in the CDM by the "system" element specified below. Here "system" is used in its most generic sense, meaning any component or item in the equipment hierarchy. A "system" could represent the vehicle, an aircraft system, subsystem, or subassembly.

At any level in this hierarchy (vehicle, system, or subassembly), elements may reference associated information. They may reference procedural task information ("task"), descriptive information ("desc"), parts information ("partinfo"), fault information ("faultinf"), or operational information ("operinfo"). This information should be attached to the level where it is most appropriate. For example, vehicle towing procedures should be referenced at the "vehicle" level. The removal task for a circuit card in the radar should be referenced at the radar "system" level.

The "system" element, like most elements in the CDM, have a set of attributes ("name", "type", "itemid", and "xrefs") which describe the nature of the element's content. These include attributes defining the element's "name" and "type". There is also an "itemid" attribute used to indicate which piece of equipment (or "item") is related to the information element. The "itemid" could be a reference designator, a SSSN number, a part number, etc., depending on the item of interest. There is also an attribute "xref" which defines relational links between the element and other elements in the database. An "xref" specifies the reference identifier for a related element and the type of relation being specified.

```
******************************************************************--
>

<!ELEMENT system     - o  ( context* ) >
<!ATTLIST system     %ids;
                     name       CDATA      #IMPLIED
                     type       CDATA      #IMPLIED
                     itemid     CDATA      #IMPLIED
                     xref       IDREFS     #IMPLIED
                     system     %refids;   #IMPLIED
                     operinfo   %refids;   #IMPLIED
                     descinfo   %refids;   #IMPLIED
                     task       %refids;   #IMPLIED
                     partinfo   %refids;   #IMPLIED
```

```
                        faultinf  %refids;  #IMPLIED>

<!-- *****************************************************************

        Operational Information

**********************************************************************-
->

<!ELEMENT operinfo - o  ( context*) >
<!ATTLIST operinfo %ids;
                        name      CDATA     #IMPLIED
                        type      CDATA     #IMPLIED
                        itemid    CDATA     #IMPLIED
                        xref      IDREFS    #IMPLIED
                        descinfo  %refids;  #IMPLIED
                        task      %refids;  #IMPLIED >
```

```
<!--******************************************************************
```

Descriptive Information

The element "descinfo" is used to define general purpose, non-procedural, narrative information such as theory of operation, schematics, wiring diagrams, etc. "Descinfo" is a very flexible, general purpose information node. It can be used to describe any arbitrary, hierarchical hypertext-like node containing sub-paragraphs ("descinfo"), data ("title", "text", "table", "graphic", "annot", "audio", "video", "process"), user interaction instructions ("prompt"), and assertion properties ("assertion") which are asserted whenever the "descinfo" is read.

```
*****************************************************************-->

<!ELEMENT descinfo - o   ( context* )>
<!ATTLIST descinfo %ids;
                    name       CDATA      #IMPLIED
                    type       CDATA      #IMPLIED
                    itemid     CDATA      #IMPLIED
                    xref       IDREFS     #IMPLIED
                    assertion  IDREFS     #IMPLIED
                    descinfo   %refids;   #IMPLIED
                    title      %refid;    #IMPLIED
                    text       %refid;    #IMPLIED
                    table      %refids;   #IMPLIED
                    graphic    %refids;   #IMPLIED
                    audio      %refids;   #IMPLIED
                    video      %refids;   #IMPLIED
                    process    %refids;   #IMPLIED
                    annot      %refids;   #IMPLIED
                    prompt     %refids;   #IMPLIED>
```

```
<!--************************************************************

          Task Information

The elements "task," "input," and "step" define a maintenance task
or procedure.  A "task" consists of a set of input conditions
("input"), a list of steps ("step"), a list of follow-on tasks
("followon"), and the attributes estimated time ("esttime") and
action describing "verb." The elements "reqcond" (required
conditions), "person" (personnel required), "equip" (equipment
required), "consum" (consumables), and "verb" describe the input
conditions referred to by "input" and "step."  The additional
elements "warning," "caution," "note," and "annot" (user
annotations) are referenced by "input" and "step."

**************************************************************-->


<!ELEMENT task - o  ( context* )>
<!ATTLIST task %ids;
                    name       CDATA      #IMPLIED
                    type       CDATA      #IMPLIED
                    itemid     CDATA      #IMPLIED
                    xref       IDREFS     #IMPLIED
                    esttime    NUTOKENS   #IMPLIED
                    verb       %refids;   #IMPLIED
                    input      %refid;    #IMPLIED
                    step       %refids;   #REQUIRED
                    followon   %refids;   #IMPLIED>


<!ELEMENT input     - o  ( context* )>
<!ATTLIST input     %ids;
                    name       CDATA      #IMPLIED
                    type       CDATA      #IMPLIED
                    itemid     CDATA      #IMPLIED
                    xref       IDREFS     #IMPLIED
                    reqcond    %refids;   #IMPLIED
                    person     %refids;   #IMPLIED
                    equip      %refids;   #IMPLIED
                    consum     %refids;   #IMPLIED
                    warning    %refids;   #IMPLIED
                    caution    %refids;   #IMPLIED
                    note       %refids;   #IMPLIED>


<!ELEMENT step - o  ( context* )>
<!ATTLIST step      %ids;
                    name       CDATA      #IMPLIED
                    type       CDATA      #IMPLIED
                    itemid     CDATA      #IMPLIED
                    xref       IDREFS     #IMPLIED
                    esttime    NUTOKENS   #IMPLIED
```

```
                              assertion IDREFS      #IMPLIED
                              step      %refids;    #IMPLIED
                              text      %refid;     #REQUIRED
                              table     %refids;    #IMPLIED
                              graphic   %refids;    #IMPLIED
                              audio     %refids;    #IMPLIED
                              video     %refids;    #IMPLIED
                              process   %refids;    #IMPLIED
                              annot     %refids;    #IMPLIED
                              prompt    %refids;    #IMPLIED
                              warning   %refids;    #IMPLIED
                              caution   %refids;    #IMPLIED
                              note      %refids;    #IMPLIED
                              verb      %refids;    #IMPLIED
                              reqcond   %refids;    #IMPLIED
                              person    %refids;    #IMPLIED
                              equip     %refids;    #IMPLIED
                              consum    %refids;    #IMPLIED>


<!ELEMENT reqcond             - o   ( context* )>
<!ATTLIST reqcond             %ids;
                              name      CDATA       #IMPLIED
                              type      CDATA       #IMPLIED
                              itemid    CDATA       #IMPLIED
                              xref      IDREFS      #IMPLIED
                              precond   IDREFS      #IMPLIED
                              elmntref  %refids;    #IMPLIED>


<!ELEMENT person    - o   ( context* )>
<!ATTLIST person    %ids;
          name      CDATA       #IMPLIED
          type      CDATA       #IMPLIED
          itemid    CDATA       #IMPLIED
          xref      IDREFS      #IMPLIED>


<!ELEMENT equip     - o   ( context* )>
<!ATTLIST equip     %ids;
          name      CDATA       #IMPLIED
          type      CDATA       #IMPLIED
          itemid    CDATA       #IMPLIED
          xref      IDREFS      #IMPLIED
          alteqids  CDATA       #IMPLIED>


<!ELEMENT consum    - o   ( context* )>
<!ATTLIST consum    %ids;
          name      CDATA       #IMPLIED
          type      CDATA       #IMPLIED
          itemid    CDATA       #IMPLIED
          xref      IDREFS      #IMPLIED
          milspec   CDATA       #REQUIRED
          mfgcode   CDATA       #REQUIRED
          govstd      CDATA      #REQUIRED
          qty         NMTOKEN    #REQUIRED>

<!ELEMENT verb - o  ( context* )>
<!ATTLIST verb %ids;
          name        CDATA      #IMPLIED
          type        CDATA      #IMPLIED
          itemid      CDATA      #IMPLIED
          xref        IDREFS     #IMPLIED>


<!ELEMENT warning     - o  ( context* )>
<!ATTLIST warning     %ids;
          name        CDATA      #IMPLIED
          type        CDATA      #IMPLIED
          itemid      CDATA      #IMPLIED
          xref        IDREFS     #IMPLIED
          text        %refid;    #REQUIRED>

<!ELEMENT caution     - o  ( context* ) >
<!ATTLIST caution     %ids;
          name        CDATA      #IMPLIED
          type        CDATA      #IMPLIED
          itemid      CDATA      #IMPLIED
          xref        IDREFS     #IMPLIED
          text        %refid;    #REQUIRED>

<!ELEMENT note - o  ( context* ) >
<!ATTLIST note %ids;
          name        CDATA      #IMPLIED
          type        CDATA      #IMPLIED
          itemid      CDATA      #IMPLIED
          xref        IDREFS     #IMPLIED
          text        %refid;    #REQUIRED>

<!ELEMENT annot       - o  ( context* ) >
<!ATTLIST annot       %ids;
          name        CDATA      #IMPLIED
          type        CDATA      #IMPLIED
          itemid      CDATA      #IMPLIED
          xref        IDREFS     #IMPLIED
          text        %refid;    #REQUIRED
          user        CDATA      #IMPLIED>
```

```
<!--**********************************************************

          Parts Information

The elements "partinfo" and "partbase" define detailed parts
information.  "Partinfo" describes an item by its reference
designator ("refdes") which categorizes parts by their place in
the system-subsystem hierarchy.  "Partinfo" describes the
maintainer's view of the part information.  Each "partinfo" element
is related to a "partbase" which describes the item in terms of its
part number ("partnum").  "Partbase" describes the supply system's
view of the part information.  Several "partinfo" items could be
related to the same "partbase."

*****************************************************************-->


<!ELEMENT partinfo   - o   ( context* )>
<!ATTLIST partinfo   %ids;
          name       CDATA      #IMPLIED
          type       CDATA      #IMPLIED
          itemid     CDATA      #IMPLIED
          xref       IDREFS     #IMPLIED
          partbase   IDREFS     #REQUIRED
          refdes     NMTOKEN    #REQUIRED
          nounid     NUTOKEN    #IMPLIED
          nountype   NUTOKEN    #IMPLIED
          unitsper   NUTOKEN    #REQUIRED
          indxnum    NUTOKEN    #REQUIRED
          usablon    NUTOKEN    #REQUIRED
          mtbf       CDATA      #REQUIRED
          replvl     CDATA      #IMPLIED
          graphic    %refids;   #REQUIRED>

<!ELEMENT partbase   - o  EMPTY >
<!ATTLIST partbase   %ids;
          name       CDATA      #IMPLIED
          type       CDATA      #IMPLIED
          itemid     CDATA      #IMPLIED
          xref       IDREFS     #IMPLIED
          partnum    CDATA      #REQUIRED
          cage       CDATA      #REQUIRED
          smr        CDATA      #REQUIRED
          hci        CDATA      #REQUIRED>
```

<!--***

Fault Information

Three types of fault information can be described in the CDM: (A) fault reporting decision trees, (B) fault isolation decision trees, and (C) dynamic fault isolation models (such as AFHRL's MDAS model). The fault reporting and isolation decision trees are static, predefined decision sequences. A dynamic fault model generates the decision sequence at display time from a fault model of the equipment. In the case of a decision tree, the complete tree is defined in the data. In the case of a dynamic fault isolation model, only the data needed to represent the fault model of the equipment is defined in the data.

Any of these diagnostic data structures can be described in terms of diagnostic tests ("test"), test outcomes ("outcome"), fault states ("fltstate"), repairable faults ("fault"), and fault rectification actions ("rect"). The general logic is that you begin a fault reporting or isolation process with a "test", which may be as simple as "what symptoms did you observe?", or as a complex as a 50-step checkout procedure. Each "test" will have associated "outcomes" which associate possible test results with new fault states ("fltstate"). Test results are described as "precond" statements (e.g., voltage = 4.5v., light = dim, faultcode = A123, ...) which are asserted by the "test" procedure as the test is performed. The "outcome" elements relate those possible test results to fault states.

A "fltstate" state represents a node in a fault isolation decision tree or a set of plausible faults in a dynamic fault model . A "fltstate"state provides the information necessary to select the next diagnostic test. In a decision tree, the test is explicitly identified by the "test" attribute of "fltstate." In a dynamic fault model, the "test" is not explicitly identified (i.e., the "test" attribute is empty), and the "fltstate" specifies a list of implicated faults ("impfault") and a list of exculpated faults ("expfault") for that state. Implicated faults are those which are suspected as being bad in the fault state. Exculpated faults are those known to be good in the fault state. These fault lists are then used by dynamic software to generate a list of appropriate "tests" which will further isolate the list of implicated faults. No matter how the test is selected, statically by the data or dynamically by the software, the selected "test" is performed and the process continues until a fault is isolated. In a decision tree, a fault is identified when you reach a final "fltstate" node which does not reference a "test", but lists the identified "fault." In a dynamic fault model, the final fault is identified by the software and is not explicitly represented in the "fltstate" element.

Once a "fault" is identified, the rectification (i.e., repair) procedure ("rect") associated with the "fault" is performed. Rectification actions also have an associate "test" which is generally a checkout task to verify that the rectification action successfully fixed the problem. The "rect" element also has a fault attribute which is a list of faults that identifies all of the faults repaired by the rectification action.

Tests and rectifications can be performed by a human or machine agent. The elements "test" and "rect" have an "agent" attribute which states whether the action is performed by a human or a machine.

```
******************************************************************-->


<!ELEMENT faultinf  - o   ( context* )>
<!ATTLIST faultinf  %ids;
          name      CDATA      #IMPLIED
          type      CDATA      #IMPLIED
          itemid    CDATA      #IMPLIED
          xref      IDREFS     #IMPLIED
          test      %refids;   #REQUIRED
          fault     %refids;   #REQUIRED>

<!--   Deleted  faultrep,  faultiso,  faultmodel,  these  can  be
represented in "type" -->


<!ELEMENT test - o  ( context* )>
<!ATTLIST test      %ids;
          name      CDATA      #IMPLIED
          type      CDATA      #IMPLIED
          itemid    CDATA      #IMPLIED
          xref      IDREFS     #IMPLIED
          text      %refid;    #IMPLIED
          agent     ( human | machine ) "human"
          task      %refids;   #REQUIRED
          outcome   %refids;   #REQUIRED
          range     CDATA      #IMPLIED>


<!ELEMENT outcome   - o   ( context* ) >
<!ATTLIST outcome   %ids;
          name      CDATA      #IMPLIED
          type      CDATA      #IMPLIED
          itemid    CDATA      #IMPLIED
          xref      IDREFS     #IMPLIED
          text      %refid;    #IMPLIED
          precond   IDREFS     #REQUIRED
          fltstate  %refids;   #REQUIRED>
```

```
<!ELEMENT fltstate  - o  ( context* )>
<!ATTLIST fltstate  %ids;
          name      CDATA      #IMPLIED
          type      CDATA      #IMPLIED
          itemid    CDATA      #IMPLIED
          xref      IDREFS     #IMPLIED
          text      %refid;    #IMPLIED
          expfault  %refids;   #IMPLIED
          impfault  %refids;   #IMPLIED
          weight    NUTOKENS   #IMPLIED
          test      %refid;    #IMPLIED>

<!ELEMENT fault     - o  ( context* )>
<!ATTLIST fault     %ids;
          name      CDATA      #IMPLIED
          type      CDATA      #IMPLIED
          itemid    CDATA      #IMPLIED
          xref      IDREFS     #IMPLIED
          mtbf      CDATA      #IMPLIED
          fltstate  %refids;   #IMPLIED
          text      %refid;    #IMPLIED
          rect      %refids;   #REQUIRED
          partinfo  %refids;   #IMPLIED>

<!ELEMENT rect - o  ( context* )>
<!ATTLIST rect %ids;
          name      CDATA      #IMPLIED
          type      CDATA      #IMPLIED
          itemid    CDATA      #IMPLIED
          xref      IDREFS     #IMPLIED
          action    ( swap | maint )     "swap"
          agent     ( human | machine )  "human"
          text      %refid;    #IMPLIED
          task      %refids;   #REQUIRED
          test      %refids;   #IMPLIED
          fault     %refids;   #IMPLIED>
```

```
<!--*************************************************************************
```

Text Elements

"Text" is the primitive text element referenced by more complex data elements in the CDM. A "text" unit is basically a text string of "parsable character data" or PCDATA. Within a text string, attribute values ("attvalue") of other CDM elements may be referenced and inserted as text string. For example, the string may contain a reference to a standard system name, or a standard part nomenclature, or a standard task name. "Attvalue" may be used to embed one of these references in a string which tells the display system to find the value of the referenced attribute and place that value into the text string for display. By using this mechanism, standard terminology can be referenced consistently throughout the data base, and any changes to the standard terminology can be made in one location and automatically updated throughout the data base.

```
*************************************************************************-
-->

<!ELEMENT text - -   ( context*, ( #PCDATA   |   attvalue)+ )>
<!ATTLIST text       %ids;
          name       CDATA     #IMPLIED
          type       CDATA     #IMPLIED
          itemid     CDATA     #IMPLIED
          xref       IDREFS    #IMPLIED >


<!ELEMENT title      - -  ( context*, ( #PCDATA   |   attvalue )+   )
>
<!ATTLIST title      %ids;
          name       CDATA     #IMPLIED
          type       CDATA     #IMPLIED
          itemid     CDATA     #IMPLIED
          xref       IDREFS    #IMPLIED>


<!ELEMENT dictitem   - -  ( context*)>
<!ATTLIST dictitem   %ids;
          name       CDATA     #REQUIRED
          type ( gloss | abbsym | symbol | other )      "other"
          itemid     CDATA     #IMPLIED
          xref       IDREFS    #IMPLIED
          def        %refids   #REQUIRED>


<!ELEMENT attvalue   - o   EMPTY >
<!ATTLIST attvalue   elmntref  %refid;   #REQUIRED
          attname    NAME "name">
```

```
<!--*********************************************************
```

Element Cross References

The element "xref" defines a cross reference or relational link. Each cross reference has at least one %refid which may be an internal reference (pointing to an element within a particular CDM data base) or an external reference (pointing to an element outside of the CDM). Internal references are represented by "elmntref" which is a reference id for any CDM element. External references are performed by "exrefid" which is character data describing another file or database element. All cross references may have a type ("relation") which is a text string describing the nature of the reference (e.g., "theory," "IPB," "schematic"). There is an optional attribute "attname" which may be used to narrow the "xref" to a particular attribute value of the cross-referenced element.

```
************************************************************--
>
```

```
<!ELEMENT xref - o  EMPTY >
<!ATTLIST xref id   ID   #REQUIRED
          relation  CDATA     #IMPLIED
          elmntref  %refids;  #IMPLIED
          attname   NAMES     #IMPLIED
          exrefid   CDATA     #IMPLIED >
```

```
<!--*********************************************************************

          Tables and Lists

"Table," "colhddef," and "entry" define the structure for a table
of information.  The cells or "entries" of a table may be a "text"
unit or any element identified by an "refid."

A "list" is a general purpose structure used to group individual
elements into a list of elements which share a common context.
For example, if you wanted to specify that a list of steps were
all to be performed if a certain precondition were true, you could
group those steps into a list with a single context which specified
the desired precondition.

*********************************************************************-->

<!ELEMENT table     - -  ( context*) >
<!ATTLIST table     %ids;
          name      CDATA     #IMPLIED
          type      CDATA     #IMPLIED
          itemid    CDATA     #IMPLIED
          xref      IDREFS    #IMPLIED
          colhddef  IDREFS    #REQUIRED
          entry     IDREFS    #REQUIRED


<!ELEMENT colhddef  - o  EMPTY>
<!ATTLIST colhddef  id   ID   #REQUIRED
          name      CDATA     #IMPLIED
          type      CDATA     #IMPLIED
          colnum    NUTOKEN   #REQUIRED>

<!ELEMENT entry     - o  EMPTY>
<!ATTLIST entry     id   ID   #REQUIRED
          col       NUTOKEN   #REQUIRED
          row       NUTOKEN   #REQUIRED
          text      %refid;   #IMPLIED
          elmntref  %refid;   #IMPLIED>


<!ELEMENT list - o  ( context* ) >
<!ATTLIST list %ids;
          elmntref  %refids;  #REQUIRED>
```

<!--**
*

Graphics

The CDM allows graphics to be referenced from external graphics files or embedded in the CDM data base. The element "grphprim" may contain a "file" name which identifies an external file containing a graphic data in any of the enumerated formats (cgm, iges, dxf, fax, ...). The same graphic data may also be included directly in the CDM by putting the data in the "#PCDATA " content portion of a "grphprim" element.

Both "graphic" and "graphprim" have a set of optional attributes to specify transformations (i.e., scaling, translating, rotating, clipping, etc.). A "graphic" or "grphrim" element may specify a transformation matrix ("transfrm"), a clipping "window," a pen shape ("penshape"), a pen pattern ("penpatt"), and a label ("text"). Transformations ("transfrm") are specified by a 9-number transformation matrix which specifies coordinate translation, scaling, reflection, and rotation in terms of homogeneous coordinates (see Chapter 3 of Rodgers, D.F., and Adams, J.A., "Mathematical Elements For Computer Graphics", McGraw Hill, 1976, for a complete definition of this matrix).

Composite graphics may be constructed by grouping transformed graphics ("graphic" or "grphprim") into a "graphic" element. These transformed, labelled, named, and typed graphic illustrations are then referenced by steps and paras in the CDM. A composite "graphic" may also specify a list of "focus" objects which are the subgraphics of interest in that particular illustration. This attribute could be used to specify which subgraphics in an illustration are to be highlighted, labelled, etc., by the presentation software.

"Graphic" and "grphprim" also may specify a minimum size ("minsize") required to satisfactorily display the graphic to the user. The minimum size is specified in terms of the amount of visual angle the graphic image should subtend on the eye. This will allow different display systems with different viewing distances to adjust the physical size of the graphic to provide the correct visual image as intended by the author.

```
******************************************************************-
-->


<!ELEMENT graphic   - o   ( context* )>
<!ATTLIST graphic         %ids;
          name            CDATA       #IMPLIED
          type            ( normal | locat | overlay |
                          schem | functblk | wiring |
                          engin )     #IMPLIED
          itemid          CDATA       #IMPLIED
          xref            IDREFS      #IMPLIED
          text            %refid;     #IMPLIED
          focus           %refids;    #IMPLIED
          graphic         %refids;    #REQUIRED
          transform       NUTOKENS #IMPLIED
          window          NUTOKENS #IMPLIED
          penshape        NUTOKENS #IMPLIED
          penpatt         NTOKENS  #IMPLIED
          minsize         NUTOKENS  #IMPLIED>

<!--  tgraphic was incorporated in graphic -->


<!ELEMENT grphprim  - -  ( context*, #PCDATA )>
<!ATTLIST grphprim  %ids;
          name      CDATA       #IMPLIED
          type      ( normal | locat | overlay |
                    schem | functblk | wiring |
                    engin )     #IMPLIED
          itemid    CDATA       #IMPLIED
          xref      IDREFS      #IMPLIED
          text      %refid;     #IMPLIED
          file      CDATA       #IMPLIED
          coding    (cgmchar | cgmbin | cgmclear |
                    fax | iges | dxf | gks)   "cgmbin"
          transfrm  NUTOKENS    #IMPLIED
          window    NUTOKENS    #IMPLIED
          penshape  CDATA       #IMPLIED
          penpatt   CDATA       #IMPLIED
          minsize   NUTOKENS    #IMPLIED>
```

```
<!--*********************************************************

          Audio - Video - Process

The elements "audio", "video", and "process" are references to
either a file name or an external source which contains an audio
sequence, a video sequence, or a software process, respectively.

**************************************************************-->

<!ELEMENT audio      - o   ( context* ) >
<!ATTLIST audio      %ids;
          name       CDATA      #IMPLIED
          type       CDATA      #IMPLIED
          itemid     CDATA      #IMPLIED
          xref       IDREFS     #IMPLIED
          file       CDATA      #IMPLIED
          exrefid    CDATA      #IMPLIED >

<!ELEMENT video      - o   ( context* ) >
<!ATTLIST video      %ids;
          name       CDATA      #IMPLIED
          type       CDATA      #IMPLIED
          itemid     CDATA      #IMPLIED
          xref       IDREFS     #IMPLIED
          file       CDATA      #IMPLIED
          exrefid    CDATA      #IMPLIED >

<!ELEMENT process    - o   ( context* ) >
<!ATTLIST process    %ids;
          name       CDATA      #IMPLIED
          type       CDATA      #IMPLIED
          itemid     CDATA      #IMPLIED
          xref       IDREFS     #IMPLIED
          file       CDATA      #IMPLIED
          exrefid    CDATA      #IMPLIED >
```

<!--***

Prompts

A "prompt" specifies either a fill-in-the-blank ("fillin") or menu choice ("menu") question for the user. Prompts are characterized in terms of property-value pairs (like assertions and preconditions). Basically, each prompt is associated with a "property" which specifies the property which will be asserted along with the user's response when the prompt is answered. If the prompt is a "fillin" the user's response will be asserted as the "value" of the specified "property". If the prompt is a "menu", the user's "choice" selection from the menu will have an associated "value" which will be asserted as the "value" of the prompt's "property". Once this assertion is made, other elements in the system may use the information to test preconditions ("precond") concerning the asserted property.

The "text" of a prompt is the question which will be displayed to the user. The "text" of a "choice" is the menu choice which will be displayed to the user as his list of possible menu selections.

Both "fillin" and "menu" prompts can have a "default" value. In the case of a "fillin," the "default" is a text string ("CDATA") which will be used as the initial entry in the fill-in-the-blank form. In the case of a "menu", the default will be an IDREF(S) to one of the possible "choice" responses.

***-->

```
<!ELEMENT prompt    - o  ( context* )>
<!ATTLIST prompt    %ids;
          name      CDATA      #IMPLIED
          type      CDATA      #IMPLIED
          itemid    CDATA      #IMPLIED
          xref      IDREFS     #IMPLIED
          text      %refid;    #IMPLIED
          fillin    %refids;   #IMPLIED
          menu      %refids;   #IMPLIED

<!ELEMENT fillin    - o  ( context* ) >
<!ATTLIST fillin    %ids;
          name      CDATA      #IMPLIED
          type      CDATA      #IMPLIED
          itemid    CDATA      #IMPLIED
          xref      IDREFS     #IMPLIED
          text      %refid;    #REQUIRED
          property  IDREF      #REQUIRED
          range     CDATA      #IMPLIED
          default   CDATA      #IMPLIED>
```

```
<!ELEMENT menu - o  ( context* ) >
<!ATTLIST menu      %ids;
          name      CDATA       #IMPLIED
          type      CDATA       #IMPLIED
          itemid    CDATA       #IMPLIED
          xref      IDREFS      #IMPLIED
          text      %refid;     #REQUIRED
          property  IDREF       #REQUIRED
          select    ( single | multiple )     "single"
          choice    IDREFS      #REQUIRED
          default   IDREFS      #IMPLIED>


<!ELEMENT choice    - o  EMPTY >
<!ATTLIST choice    id   ID     #REQUIRED
          text      %refid;     #REQUIRED
          value     IDREFS      #REQUIRED>
```

```
<!--**************************************************************
```

Context and Assertions

Every CDM composite object also has context. Vehicle configuration, security level, and technician skill level are examples of context properties which determine the applicability of a particular data element to the situation at hand. "Context" consists of a set of frequently used "effectivity" attributes (security, config, track, version) , and a list of user-defined "precond" (preconditions).

"Precond" and "assertion" are both defined in terms of property-value pairs. A "property" is any "text" string which defines a property. A "value" is another "text" string defining the value. Property-value pairs may be asserted or tested by the run-time presentation software. An "assertion" on a paragraph or step will be asserted whenever that paragraph or step is performed. A "precond" is a test of a property previously asserted. The property element also has an "elmntref" attribute which is an optional attribute which may be used to indicate a prompt or task which can be activated to acquire a value for the property if none has been asserted.

```
*****************************************************************-
->

<!ELEMENT context    - o   EMPTY>
<!ATTLIST context    id    ID        #REQUIRED
          security   (uc | c | s | ts)      #IMPLIED
          restrict   NMTOKENS  #IMPLIED
          release    NMTOKENS  #IMPLIED
          codeword   NMTOKENS  #IMPLIED
          scilevel   NUMBER    #IMPLIED
          diglyph    NMTOKENS  #IMPLIED
          config     NMTOKENS  #IMPLIED
          maintlvl   CDATA     #IMPLIED
          track      NUTOKENS  #IMPLIED
          version    NUTOKENS  #IMPLIED
          valstat    CDATA     #IMPLIED
          verstat    CDATA     #IMPLIED
          precond    IDREFS    #IMPLIED>


<!ELEMENT assertion - o    EMPTY>
<!ATTLIST assertion id     ID       #REQUIRED
          polarity         ( pos | neg )  "pos"
          property         IDREF        #REQUIRED
          value            IDREFS       #REQUIRED>
```

```
<!ELEMENT precond   - o   EMPTY >
<!ATTLIST precond   id    ID    #REQUIRED
          polarity        ( pos | neg )  "pos"
          property        IDREF      #REQUIRED
          op        ( eq | lt | lte | gt | gte | in )   "eq"
          value           IDREFS     #REQUIRED>

<!ELEMENT property   -  o EMPTY>
<!ATTLIST property  id      ID      #REQUIRED
          text      %refid;   #REQUIRED
          elmntref  %refid;   #IMPLIED>

<!ELEMENT value     - o   EMPTY>
<!ATTLIST value     id      ID      #REQUIRED
          text      %refid;   #REQUIRED>


]>
```

APPENDIX B

JOB GUIDE DOCUMENT TYPE DEFINITION

```
<!DOCTYPE jobguide [

<!ENTITY % yesorno "NUMBER">

<!ENTITY % bodyatt "id  ID  #IMPLIED
          inschlvl NUTOKEN #IMPLIED
          delchlvl NUTOKEN #IMPLIED
          label NMTOKEN #IMPLIED
          texttype  NUMBER  #IMPLIED
          itemid  NMTOKEN #IMPLIED
          config NUTOKENS #IMPLIED
          skilltrk CDATA #IMPLIED
          hcp %yesorno; '0'
          xref IDREF #IMPLIED" >

<!ENTITY % change "stchg | endchg" >

<!ENTITY % colfmt "colnum  NUMBER #REQUIRED
          datatype CDATA #IMPLIED
          left %yesorno; '0'
          right %yesorno; '0'
          center %yesorno; '0'
          char CDATA #IMPLIED
          width  NUMBER #IMPLIED
          leader CDATA #IMPLIED
          colsep %yesorno; #IMPLIED
          rowhead %yesorno; '0' " >

<!ENTITY % contdef "TBD" >

<!ENTITY % def "" >

<!ENTITY % asyntxt "steng | endemg | %change; | stemph | endemph"
>

<!ENTITY % list "seqlist | randlist | deflist" >

<!ENTITY % nums "partno | modelno | sssn | figindex" >

<!ENTITY % secur "security  (uc | c | s | ts)  #IMPLIED
          restrict NMTOKENS #IMPLIED
          release NMTOKENS #IMPLIED
          codeword NMTOKENS #IMPLIED
          scilevel %yesorno '0'
          diglyph NMTOKENS #IMPLIED" >

<!ENTITY % spcpara "(warning*, caution*, note*)" >
```

```
<!ENTITY % text "(#PCDATA | %asyntxt; | %nums; | tool | testeq |
material |
 torquevl | xref | graphic | subscrpt | supscrpt)*" >

<!NOTATION iges PUBLIC "-//USA-DOD//NOTATION Initial Graphics
Exchange
Specification//EN" >

<!NOTATION cgm PUBLIC "-//USA-DOD//NOTATION Computer Graphics
Metafile//EN"

>
<!NOTATION fax PUBLIC "-//USA-DOD//NOTATION CCITT Group 4
Facsimile//EN"

>
<!NOTATION math-eqn PUBLIC "-//USA-DOD//NOTATION EQN Mathematical

Notation//EN" >

<!NOTATION math-tex PUBLIC "-//USA-DOD//NOTATION TEX Mathematical

Notation//EN" >

<!NOTATION mathsmff PUBLIC "-//USA-DOD//NOTATION IBM Scientific and

Mathematical Formula
Format//EN" >

<!ENTITY % ISOlat1 PUBLIC "ISO 8879-1986//ENTITIES Added Latin
1//EN"

>
<!ENTITY % ISOlat2  "ISO 8879-1986//ENTITIES Added Latin 2//EN" >


<!ENTITY % ISOgrk1  "ISO 8879-1986//ENTITIES Greek Letters//EN" >


<!ENTITY % ISOnum  "ISO 8879-1986//ENTITIES Numeric and Special
Graphic//EN"

>
<!ENTITY % ISOdia  "ISO 8879-1986//ENTITIES Diacritical Marks//EN"


>
<!ENTITY % ISOpub  "ISO 8879-1986//ENTITIES Publishing//EN" >
```

```
<!ENTITY % ISObox "ISO 8879-1986//ENTITIES Box and Line
Drawing//EN"

>
<!ENTITY % ISOtech "ISO 8879-1986//ENTITIES General Technical//EN"


>
<!ENTITY % ISOamso "ISO 8879-1986//ENTITIES Added Math Symbols:

Ordinary//EN" >

<!ENTITY % ISOamsb "ISO 8879-1986//ENTITIES Added Math Symbols:

Binary Operators//EN" >

<!ENTITY % ISOamsr "ISO 8879-1986//ENTITIES Added Math Symbols:

Relations//EN" >

<!ENTITY % ISOamsn "ISO 8879-1986//ENTITIES Added Math Symbols:

Negated Relations//EN" >

<!ENTITY % ISOamsa "ISO 8879-1986//ENTITIES Added Math Symbols:

Arrow Relations//EN" >

<!ENTITY % ISOamsc "ISO 8879-1986//ENTITIES Added Math Symbols:

Delimiters//EN" >

<!--%ISOlat1; %ISOlat2; %ISOgrk1; %ISOnum; %ISOdia; %ISOpub;
%ISObox;
%ISOtech; %ISOamso; %ISOamsb;
%ISOamsr; %ISOamsn; %ISOamsa; %ISOamsc; -->

<!ENTITY effdate "The effective date of this publication is
&effdate1;.
Instructions herein shall not be used
prior to that date." >

<!ENTITY effdate1 "Originator Supplies Effective Date Here" >

<!ENTITY disclos "If this technical manual is approved for release

to a foreign government, it is furnished
```

```
upon the condition that it will not be released to another nation

without specific authority of the &disclos1;
of the United States, that it will be used for military purposes

only, that individual or corporate rights
originating in the information, whether patented or not will be
respected,
that the recipient will report
promptly to the US, any known or suspected compromise, and that the

information will be provided substantially the same degree of
security
afforded it by the Department of Defense of the United States." >


<!ENTITY disclos1 "Originator Supplies Appropriate Department or

Agency Here" >

<!ENTITY pgclass "This publication consists of &pgclass1; secret

pages of &pgclass2; total pages (N). Copy
No. &pgclass3; of &pgclass4; copies." >

<!ENTITY pgclass1 "Originator Supplies Number of Secret Pages Here"


>
<!ENTITY pgclass2 "Originator Supplies Total Number of Pages Here"


>
<!ENTITY pgclass3 "Originator Supplies Number of This Copy Here"


>
<!ENTITY pgclass4 "Originator Supplies Total Number of Copies Here"


>
<!ENTITY fouo "For Official Use Only" >

<!ENTITY fcuo "For Consortium Use Only" >

<!ENTITY fmuo "For MAP Use Only" >

<!ENTITY distrib "This publication is required for official use or
```

```
for administrative or operational purposes
only. Distribution is limited to U.S. Government Agencies. Other

requests for this document must be
referred to &distribl;." >

<!ENTITY distribl "Originator Supplies Applicable Activity or
Address
Here" >

<!ENTITY sprsede "This manual supersedes &sprsedel; dated
&sprsede2;,
which shall be destroyed in accordance with applicable security
regulations." >

<!ENTITY sprsedel "Originator Supplies Superseded Publication
Number
Here" >

<!ENTITY sprsede2 "Originator Supplies Supersedure Date Here" >

<!ENTITY sprpre "This manual supersedes preliminary &sprprel; date

&sprpre2;." >

<!ENTITY sprprel "Originator Supplies Superseded Preliminary
Publication
Number Here" >

<!ENTITY sprpre2 "Originator Supplies Supersedure Date Here" >


<!ELEMENT jobguide - - (front, body) >
<!ATTLIST jobguide status  (revision | change | prelim | draft |

formal)  "formal"
                revno  NUMBER  "0"
                chgno  NUMBER  "0"
                %secur;       >

<!ELEMENT front - - (idinfo, lep, verstat?, contents?, intro?) >

<!ATTLIST front %secur;       >

<!ELEMENT idinfo - - (pubno+, prepubno?, %spcpara;, titleblk, (mfr,

contrno)+,docmfr?, notice+, pubdate,
chgdate?)  >
<!ATTLIST idinfo %secur;     >

<!ELEMENT pubno - - (user, docno) >
<!ATTLIST pubno %secur;     >

<!ELEMENT prepubno - - (user, docno) >
<!ATTLIST prepubno %secur;     >

<!ELEMENT user - - (#PCDATA)>
<!ATTLIST user %secur;     >

<!ELEMENT titleblk - - (doctype, maintlvl*, sssnrng, prtitle) >
<!ATTLIST titleblk %secur;     >

<!ELEMENT doctype - - (%text;)>
<!ATTLIST doctype %secur;     >

<!ELEMENT maintlvl - - (%text;) >
<!ATTLIST maintlvl %secur;     >

<!ELEMENT sssnrng - - (%text;) >
<!ATTLIST sssnrng %secur;     >

<!ELEMENT prtitle - - (nomen, type?) >
<!ATTLIST prtitle %secur;     >

<!ELEMENT nomen - - (#PCDATA) >
<!ATTLIST nomen %secur;     >

<!ELEMENT type - - (#PCDATA) >
<!ATTLIST type %secur;     >

<!ELEMENT mfr - - (#PCDATA) >
<!ATTLIST mfr %secur;     >

<!ELEMENT docmfr - - (#PCDATA) >
<!ATTLIST docmfr %secur;     >

<!ELEMENT contrno - - (#PCDATA) >
<!ATTLIST contrno %secur;     >

<!ELEMENT notice - - ((%text;) | (%list;))* >
<!ATTLIST notice type  (dist | auth | offic | branch | pgclass |

discl | supersed |effdate | suppl | nopg |
noclaspg)   #IMPLIED
            %secur;     >
```

```
<!ELEMENT pubdate - - (#PCDATA) >
<!ATTLIST pubdate %secur;      >

<!ELEMENT chgdate - - (#PCDATA) >
<!ATTLIST chgdate %secur;      >

<!ELEMENT date - - (#PCDATA) >
<!ATTLIST date %secur;      >

<!ELEMENT lep - o EMPTY >

<!ELEMENT verstat - - (para, (para | %list; | (%spcpara;))*) >
<!ATTLIST verstat %secur;      >

<!ELEMENT contents - o EMPTY >

<!ELEMENT intro - - (title, (title, (paratext | abbrsect | symsect

| tctorec | (%list;))+)+) >
<!ATTLIST intro %secur;      >

<!ELEMENT seqlist - - (title?, item*) >
<!ATTLIST seqlist label  CDATA  #IMPLIED
                  %secur;     >

<!ELEMENT randlist - - (title?, item*) >
<!ATTLIST randlist %secur;      >

<!ELEMENT deflist - - (term, def)+ >
<!ATTLIST deflist %secur;      >

<!ELEMENT item - - (%text;) >
<!ATTLIST item  %bodyatt;
                  %secur;     >

<!ELEMENT symsect - - (deflist) >
<!ATTLIST symsect %secur;      >

<!ELEMENT abbrsect - - (deflist) >
<!ATTLIST abbrsect %secur;      >

<!ELEMENT term - - (%text;) >
<!ATTLIST term %secur;      >

<!ELEMENT tctorec - - (pubno, title, date?)+ >
<!ATTLIST tctorec %secur;      >
```

```
<!ELEMENT title - - (%text;) >
<!ATTLIST title %secur;
                texttype  NUMBER  #IMPLIED >

<!ELEMENT def - - (%text; | table)+     -(%list;)>
<!ATTLIST def %secur;      >

<!ELEMENT paratext - - (%text; | (%list;))* >
<!ATTLIST paratext %secur;
                texttype  NUMBER  #IMPLIED >

<!ELEMENT para - - (%text; | %list; | (%spcpara;))+ +(figure |
chart | table)>
<!ATTLIST para %bodyatt;
                %sacur;      >

<!ELEMENT chart - - (title, (tabdef | stdtable), thead?, tbody)

>
<!ATTLIST chart %bodyatt;
                %secur;      >

<!ELEMENT table - - (title, (tabdef | stdtable), thead?, tbody)

>
<!ATTLIST table %bodyatt;
                %secur;      >

<!ELEMENT stdtable - o EMPTY >
<!ATTLIST stdtable name  ENTITY  #REQUIRED
                %secur;      >

<!ELEMENT tabdef - - (colbddef)* >
<!ATTLIST tabdef siderule  %yesorno;  "0"
                colsep  %yesorno;  "0"
                rowsep  %yesorno;  "0"
                orient  (port | land)  "port"
                width  (page | column)  "page"
                %secur;      >

<!ELEMENT colbddef - o EMPTY >
<!ATTLIST colbddef %colfmt;      >

<!ELEMENT thead - - (row+) >
<!ATTLIST thead %secur;      >
```

```
<!ELEMENT tbody - - (row+) >
<!ATTLIST tbody %secur;     >

<!ELEMENT row - - (entry+) >
<!ATTLIST row number  NUMBER  #REQUIRED
                rowsep  %yesorno;  #IMPLIED
                %secur;     >

<!ELEMENT entry - - (%text;)        -(figure | table | chart)>
<!ATTLIST entry col  NUTOKEN  #REQUIRED
                row  NMTOKEN  #CURRENT
                rotate  NUMBER  "0"
                %secur;     >

<!ELEMENT figure - - (graphic+, title, legend?) >
<!ATTLIST figure %bodyatt;
                %secur;     >

<!ELEMENT legend - - (deflist) >
<!ATTLIST legend %secur;     >

<!ELEMENT body - - (section1*, section+) +(figure | table | chart)>

<!ATTLIST body %secur;     >

<!ELEMENT section1 - - (title, (title?, (paratext |(%spcpara;) |

%list;)+)+)  +(figure | table | chart)>
<!ATTLIST section1 %bodyatt;
                %secur;     >

<!ELEMENT  section  -  -  (title,  incond,  taskinfo+,endcond*,
followon*)
                              +(figure | table | chart)>
<!ATTLIST section %bodyatt;
                  %secur;
               sssn      CDATA          #REQUIRED      >

<!ELEMENT  incond  -  -  (applic,  reqdcond?,  persreq?,  sppteqpt?,
spptdata?,
supplies?, safecond?,clreq?)  >
<!ATTLIST incond %bodyatt;
                %secur;     >

<!ELEMENT applic - - (paratext+, config*) >
<!ATTLIST applic %bodyatt;
                %secur;     >
```

```
<!ELEMENT reqdcond - - (paratext)* >
<!ATTLIST reqdcond %bodyatt;
                   %secur;     >

<!ELEMENT persreq - - (paratext*) >
<!ATTLIST persreq %bodyatt;
                   %secur;     >

<!ELEMENT sppteqpt - - (paratext)* >
<!ATTLIST sppteqpt %bodyatt;
                   %secur;     >

<!ELEMENT spptdata - - (paratext)* >
<!ATTLIST spptdata %bodyatt;
                   %secur;     >

<!ELEMENT supplies - - (paratext* | table) >
<!ATTLIST supplies %bodyatt;
                   %secur;     >

<!ELEMENT safecond - - ((%spcpara;), paratext*) >
<!ATTLIST safecond %bodyatt;
                   %secur;     >

<!ELEMENT clreq - - (paratext*) >
<!ATTLIST clreq %bodyatt;
                   %secur;     >

<!ELEMENT taskinfo - - (title, (stepgrp | step1)*) >
<!ATTLIST taskinfo %bodyatt;
                   %secur;     >

<!ELEMENT stepgrp - - (%spcpara, step1, step1*) >
<!ATTLIST stepgrp %bodyatt;
                   %secur;     >

<!ELEMENT step1 - - ((%spcpara;), paratext+, (step2)*) >
<!ATTLIST step1 %bodyatt;
                   %secur;
                   ipi  NAME  #IMPLIED
                   person  NAME  #IMPLIED >

<!ELEMENT step2 - - ((%spcpara;), paratext+, (step3)*) >
<!ATTLIST step2 %bodyatt;
                   %secur;
                   ipi  NAME  #IMPLIED
```

```
                    person  NAME  #IMPLIED >

<!ELEMENT step3 - - ((%spcpara;), paratext+) >
<!ATTLIST step3 %bodyatt;
                %secur;
                ipi  NAME  #IMPLIED
                person  NAME  #IMPLIED >

<!ELEMENT endcond - - (paratext*) >
<!ATTLIST endcond %secur;      >

<!ELEMENT followon - - (paratext*, (%spcpara;)) >
<!ATTLIST followon %secur;     >

<!ELEMENT figindex - - (#PCDATA) >
<!ATTLIST figindex %secur;     >

<!ELEMENT docno - - (#PCDATA) >
<!ATTLIST docno %secur;      >

<!ELEMENT sssn - - (#PCDATA) >
<!ATTLIST sssn %secur;      >

<!ELEMENT tool - - (#PCDATA) >
<!ATTLIST tool %secur;      >

<!ELEMENT testeq - - (#PCDATA) >
<!ATTLIST testeq %secur;      >

<!ELEMENT material - - (#PCDATA) >
<!ATTLIST material %secur;     >

<!ELEMENT torquevl - - (#PCDATA) >
<!ATTLIST torquevl %secur;     >

<!ELEMENT stchg - o EMPTY >
<!ATTLIST stchg level  NUMBER  #IMPLIED
                status  (add | delete)  #IMPLIED
                %secur;     >

<!ELEMENT endchg - o EMPTY >
<!ATTLIST endchg level  NUMBER  #IMPLIED
                 status  (add | delete)  #IMPLIED
                 %secur;     >

<!ELEMENT subscrpt - - (#PCDATA, subscrpt?, supscrpt?) >
<!ATTLIST subscrpt %secur;     >
```

```
<!ELEMENT supscrpt - - (#PCDATA, subscrpt?, supscrpt?) >
<!ATTLIST supscrpt %secur;     >

<!-- 8/4/89 config changed here - now same as fault -->
<!ELEMENT config - - (#PCDATA) >
<!ATTLIST config code  CDATA #REQUIRED
                %secur; >

<!ELEMENT xref - o EMPTY >
<!ATTLIST xref xref  CDATA  #REQUIRED
               config NUTOKENS #IMPLIED
                %secur;     >

<!ELEMENT graphic - o EMPTY >
<!ATTLIST graphic boardno  CDATA  #REQUIRED
                width  NUTOKEN  #IMPLIED
                depth  NUTOKEN  #IMPLIED
                window  CDATA  #IMPLIED
                hscale  NUTOKEN  #IMPLIED
                vscale  NUTOKEN  #IMPLIED
                text  CDATA  #IMPLIED
                rotation  NUMBER  "0"
                %secur;     >

<!ELEMENT partno - - (#PCDATA) >
<!ATTLIST partno %secur;     >

<!ELEMENT modelno - - (#PCDATA) >
<!ATTLIST modelno %secur;     >

<!ELEMENT warning - - (graphic?, (paratext*)) >
<!ATTLIST warning name  ENTITY  #CONREF
                type  NAME  #IMPLIED
                xref  IDREF  #IMPLIED
                vital  (y | n)  "n"
                %secur;     >

<!ELEMENT caution - - (graphic?, (paratext*)) >
<!ATTLIST caution name  ENTITY  #CONREF
                type  NAME  #IMPLIED
                xref  IDREF  #IMPLIED
                %secur;     >

<!ELEMENT note - - (graphic?, (paratext*)) >
<!ATTLIST note name  ENTITY  #CONREF
                type  NAME  #IMPLIED
```

```
                 xref  IDREF  #IMPLIED
                 %secur;     >

<!ELEMFNT stemg - o EMPTY >

<!ELEMENT endemg - o EMPTY >

<!ELEMENT stemph - o EMPTY >
<!ATTLIST stemph type  NUMBER  #REQUIRED >

<!ELEMENT endemph - o EMPTY >
<!ATTLIST endemph type  NUMBER  #REQUIRED >


]>
```

APPENDIX C

FAULT ISOLATION MANUAL DOCUMENT TYPE DEFINITION

```
<!DOCTYPE         faultiso     [

<!ENTITY     % yesorno    "NUMBER"  >
<!ENTITY     % change     "stchg | endchg"  >
<!ENTITY     % contdef    "TBD"  >
<!ENTITY     % def   ""  >
<!ENTITY     % asyntxt    "stemg | endemg | %change; | stemph |
endemph |
 config"  >
<!ENTITY     % list  "seqlist | randlist | deflist"  >
<!ENTITY     % nums  "partno | modelno | sssn | figindex"  >
<!ENTITY     % spcpara    "(warning*, caution*, note*)"  >
<!ENTITY     % text  " %asyntxt; | %nums; | tool | testeq | material
|
 torquevl | xref | graphic |
 subscrpt | supscrpt"  >

<!ELEMENT    faultiso     - -     (front, body)  +(%text;)>
<!ATTLIST    faultiso     status        (revision | change |
                          prelim | draft | formal)     "formal"
                 revno     NUMBER              "0"
                 chgno     NUMBER                   "0"
                 security  (uc | c | s | ts)             #IMPLIED
                 restrict  NMTOKENS                 #IMPLIED
                 release   NMTOKENS                 #IMPLIED
                 codeword  NMTOKENS                 #IMPLIED
                 scilevel  %yesorno;           "0"
                 diglyph   NMTOKENS                 #IMPLIED>

<!ELEMENT     front              - -      (idinfo, lep, verstat?,
contents?, loi, lot, intro?)>
<!ATTLIST     front        security       (uc | c | s | ts)
#IMPLIED
                 restrict  NMTOKENS                 #IMPLIED
                 release   NMTOKENS                 #IMPLIED
                 codeword  NMTOKENS                 #IMPLIED
                 scilevel  %yesorno;           "0"
                 diglyph   NMTOKENS                 #IMPLIED>
<!ELEMENT    idinfo       - -     (pubno+, prepubno?,
                          %spcpara;, titleblk,
                          (mfr, contrno+)+,
                          docmfr?, notice+,
                          pubdate, chgdate?)>
<!ATTLIST     idinfo       security       (uc | c | s | ts)
#IMPLIED
                 restrict  NMTOKENS                 #IMPLIED
                 release   NMTOKENS                 #IMPLIED
                 codeword  NMTOKENS                 #IMPLIED
                 scilevel  %yesorno;           "0"
                 diglyph   NMTOKENS                 #IMPLIED>
<!ELEMENT    pubno        - -     (user, docno)>
<!ATTLIST     pubno        security       (uc | c | s | ts)
#IMPLIED
                 restrict  NMTOKENS                 #IMPLIED
```

```
                release     NMTOKENS                      #IMPLIED
                codeword    NMTOKENS                      #IMPLIED
                scilevel    %yesorno;            "0"
                diglyph     NMTOKENS                      #IMPLIED>
<!ELEMENT   prepubno          - -      (user, docno)>
<!ATTLIST    prepubno       security          (uc | c | s | ts)
#IMPLIED
                restrict    NMTOKENS                      #IMPLIED
                release     NMTOKENS                      #IMPLIED
                codeword    NMTOKENS                      #IMPLIED
                scilevel    %yesorno;            "0"
                diglyph     NMTOKENS                      #IMPLIED>
<!ELEMENT   user              - -      (#PCDATA)>
<!ATTLIST    user           security          (uc | c | s | ts)
#IMPLIED
                restrict    NMTOKENS                      #IMPLIED
                release     NMTOKENS                      #IMPLIED
                codeword    NMTOKENS                      #IMPLIED
                scilevel    %yesorno;            "0"
                diglyph     NMTOKENS                      #IMPLIED>
<!ELEMENT   titleblk          - -      (doctype, maintlvl*,
                            sssnrng, prtitle)>
<!ATTLIST    titleblk       security          (uc | c | s | ts)
#IMPLIED
                restrict    NMTOKENS                      #IMPLIED
                release     NMTOKENS                      #IMPLIED
                codeword    NMTOKENS                      #IMPLIED
                scilevel    %yesorno;            "0"
                diglyph     NMTOKENS                      #IMPLIED>
<!ELEMENT   doctype           - -      (#PCDATA)>
<!ATTLIST    doctype        security          (uc | c | s | ts)
#IMPLIED
                restrict    NMTOKENS                      #IMPLIED
                release     NMTOKENS                      #IMPLIED
                codeword    NMTOKENS                      #IMPLIED
                scilevel    %yesorno;            "0"
                diglyph     NMTOKENS                      #IMPLIED>
<!ELEMENT   maintlvl          - -      (#PCDATA)>
<!ATTLIST    maintlvl       security          (uc | c | s | ts)
#IMPLIED
                restrict    NMTOKENS                      #IMPLIED
                release     NMTOKENS                      #IMPLIED
                codeword    NMTOKENS                      #IMPLIED
                scilevel    %yesorno;            "0"
                diglyph     NMTOKENS                      #IMPLIED>
<!ELEMENT   sssnrng           - -      (#PCDATA)>
<!ATTLIST    sssnrng        security          (uc | c | s | ts)
#IMPLIED
                restrict    NMTOKENS                      #IMPLIED
                release     NMTOKENS                      #IMPLIED
                codeword    NMTOKENS                      #IMPLIED
                scilevel    %yesorno;            "0"
                diglyph     NMTOKENS                      #IMPLIED>
<!ELEMENT   prtitle           - -      (nomen, type?)>
```

```
<!ATTLIST     prtitle        security         (uc | c | s | ts)
#IMPLIED
                   restrict    NMTOKENS                   #IMPLIED
                   release     NMTOKENS                   #IMPLIED
                   codeword    NMTOKENS                   #IMPLIED
                   scilevel    %yesorno;         "0"
                   diglyph     NMTOKENS                   #IMPLIED>
<!ELEMENT     nomen              - -      (#PCDATA)>
<!ATTLIST     nomen          security         (uc | c | s | ts)
#IMPLIED
                   restrict    NMTOKENS                   #IMPLIED
                   release     NMTOKENS                   #IMPLIED
                   codeword    NMTOKENS                   #IMPLIED
                   scilevel    %yesorno;         "0"
                   diglyph     NMTOKENS                   #IMPLIED>
<!ELEMENT     type               - -      (#PCDATA)>
<!ATTLIST     type           security         (uc | c | s |  ts)
#IMPLIED
                   restrict    NMTOKENS                   #IMPLIED
                   release     NMTOKENS                   #IMPLIED
                   codeword    NMTOKENS                   #IMPLIED
                   scilevel    %yesorno;         "0"
                   diglyph     NMTOKENS                   #IMPLIED>
<!ELEMENT     mfr                - -      (#PCDATA)>
<!ATTLIST     mfr            security         (uc | c | s |  ts)
#IMPLIED
                   restrict    NMTOKENS                   #IMPLIED
                   release     NMTOKENS                   #IMPLIED
                   codeword    NMTOKENS                   #IMPLIED
                   scilevel    %yesorno;         "0"
                   diglyph     NMTOKENS                   #IMPLIED>
<!ELEMENT     docmfr             - -      (#PCDATA)>
<!ATTLIST     docmfr         security         (uc | c | s |  ts)
#IMPLIED
                   restrict    NMTOKENS                   #IMPLIED
                   release     NMTOKENS                   #IMPLIED
                   codeword    NMTOKENS                   #IMPLIED
                   scilevel    %yesorno;         "0"
                   diglyph     NMTOKENS                   #IMPLIED>
<!ELEMENT     contrno            - -      (#PCDATA)>
<!ATTLIST     contrno        security         (uc | c | s |  ts)
#IMPLIED
                   restrict    NMTOKENS                   #IMPLIED
                   release     NMTOKENS                   #IMPLIED
                   codeword    NMTOKENS                   #IMPLIED
                   scilevel    %yesorno;         "0"
                   diglyph     NMTOKENS                   #IMPLIED>
<!ELEMENT     notice             - -      (#PCDATA)>
<!ATTLIST     notice        type            (dist | auth | offic
                        | branch | pgclass |
                        discl | supersed |
                        effdate | suppl | nopg |
                        noclaspg)                         #IMPLIED
                   security   (uc | c | s |  ts)           #IMPLIED
```

```
                    restrict    NMTOKENS                   #IMPLIED
                    release     NMTOKENS                   #IMPLIED
                    codeword    NMTOKENS                   #IMPLIED
                    scilevel    %yesorno;           "0"
                    diglyph     NMTOKENS                   #IMPLIED>
<!ELEMENT      pubdate             - -       (#PCDATA)>
<!ATTLIST       pubdate         security           (uc | c | s |   ts)
#IMPLIED
                    restrict    NMTOKENS                   #IMPLIED
                    release     NMTOKENS                   #IMPLIED
                    codeword    NMTOKENS                   #IMPLIED
                    scilevel    %yesorno;           "0"
                    diglyph     NMTOKENS                   #IMPLIED>
<!ELEMENT      chgdate             - -       (#PCDATA)>
<!ATTLIST       chgdate         security           (uc | c | s |   ts)
#IMPLIED
                    restrict    NMTOKENS                   #IMPLIED
                    release     NMTOKENS                   #IMPLIED
                    codeword    NMTOKENS                   #IMPLIED
                    scilevel    %yesorno;           "0"
                    diglyph     NMTOKENS                   #IMPLIED>
<!ELEMENT      date                - -       (#PCDATA)>
<!ATTLIST       date            security           (uc | c | s |   ts)
#IMPLIED
                    restrict    NMTOKENS                   #IMPLIED
                    release     NMTOKENS                   #IMPLIED
                    codeword    NMTOKENS                   #IMPLIED
                    scilevel    %yesorno;           "0"
                    diglyph     NMTOKENS                   #IMPLIED>
<!ELEMENT      lep                 - o      EMPTY>
<!ELEMENT      loi                 - o      EMPTY>
<!ELEMENT      lot                 - o      EMPTY>
<!ELEMENT      verstat             - -       (para, (para | %list; |
                               (%spcpara;))*)>
<!ATTLIST       verstat         security           (uc | c | s |   ts)
#IMPLIED
                    restrict    NMTOKENS                   #IMPLIED
                    release     NMTOKENS                   #IMPLIED
                    codeword    NMTOKENS                   #IMPLIED
                    scilevel    %yesorno;           "0"
                    diglyph     NMTOKENS                   #IMPLIED>
<!ELEMENT      contents            - o      EMPTY>
<!ELEMENT      intro               - -       (title, (title, (paratext
                               | abbrsect | symsect |
                               tctorec |
                               (%list;))+)+)>
<!ATTLIST       intro           security           (uc | c | s |   ts)
#IMPLIED
                    restrict    NMTOKENS                   #IMPLIED
                    release     NMTOKENS                   #IMPLIED
                    codeword    NMTOKENS                   #IMPLIED
                    scilevel    %yesorno;           "0"
                    diglyph     NMTOKENS                   #IMPLIED>
<!ELEMENT      seqlist             - -       (title?, item*)>
```

```
<!ATTLIST    seqlist     label          CDATA
#IMPLIED
                   security   (uc | c | s |   ts)      #IMPLIED
                   restrict   NMTOKENS                 #IMPLIED
                   release    NMTOKENS                 #IMPLIED
                   codeword   NMTOKENS                 #IMPLIED
                   scilevel   %yesorno;         "0"
                   diglyph    NMTOKENS                 #IMPLIED>
<!ELEMENT    randlist       - -      (title?, item*)>
<!ATTLIST     randlist     security        (uc  |  c  |  s  |   ts)
#IMPLIED
                   restrict   NMTOKENS                 #IMPLIED
                   release    NMTOKENS                 #IMPLIED
                   codeword   NMTOKENS                 #IMPLIED
                   scilevel   %yesorno;         "0"
                   diglyph    NMTOKENS                 #IMPLIED>
<!ELEMENT    deflist        - -      (term, def)+>
<!ATTLIST     deflist      security        (uc  |  c  |  s  |   ts)
#IMPLIED
                   restrict   NMTOKENS                 #IMPLIED
                   release    NMTOKENS                 #IMPLIED
                   codeword   NMTOKENS                 #IMPLIED
                   scilevel   %yesorno;         "0"
                   diglyph    NMTOKENS                 #IMPLIED>
<!ELEMENT    item           - -      (#PCDATA | %text;)+>
<!ATTLIST    item        id          ID                  #IMPLIED
                   security   (uc | c | s |   ts)      #IMPLIED
                   restrict   NMTOKENS                 #IMPLIED
                   release    NMTOKENS                 #IMPLIED
                   codeword   NMTOKENS                 #IMPLIED
                   scilevel   %yesorno;         "0"
                   diglyph    NMTOKENS                 #IMPLIED>
<!ELEMENT    symsect        - -      (deflist)>
<!ATTLIST     symsect      security        (uc  |  c  |  s  |   ts)
#IMPLIED
                   restrict   NMTOKENS                 #IMPLIED
                   release    NMTOKENS                 #IMPLIED
                   codeword   NMTOKENS                 #IMPLIED
                   scilevel   %yesorno;         "0"
                   diglyph    NMTOKENS                 #IMPLIED>
<!ELEMENT    abbrsect       - -      (deflist)>
<!ATTLIST     abbrsect     security        (uc  |  c  |  s  |   ts)
#IMPLIED
                   restrict   NMTOKENS                 #IMPLIED
                   release    NMTOKENS                 #IMPLIED
                   codeword   NMTOKENS                 #IMPLIED
                   scilevel   %yesorno;         "0"
                   diglyph    NMTOKENS                 #IMPLIED>
<!ELEMENT    term           - -      (#PCDATA)>
<!ATTLIST     term         security        (uc  |  c  |  s  |   ts)
#IMPLIED
                   restrict   NMTOKENS                 #IMPLIED
                   release    NMTOKENS                 #IMPLIED
                   codeword   NMTOKENS                 #IMPLIED
```

```
                    scilevel    %yesorno;          "0"
                    diglyph     NMTOKENS                #IMPLIED>
<!ELEMENT     tctorec          - -     (pubno, title, date?)+>
<!ATTLIST      tctorec        security         (uc | c | s |   ts)
#IMPLIED
                    restrict    NMTOKENS                #IMPLIED
                    release     NMTOKENS                #IMPLIED
                    codeword    NMTOKENS                #IMPLIED
                    scilevel    %yesorno;          "0"
                    diglyph     NMTOKENS                #IMPLIED>
<!ELEMENT     title            - -     (#PCDATA)>
<!ATTLIST      title          security         (uc | c | s |   ts)
#IMPLIED
                    restrict    NMTOKENS                #IMPLIED
                    release     NMTOKENS                #IMPLIED
                    codeword    NMTOKENS                #IMPLIED
                    scilevel    %yesorno;          "0"
                    texttype    NUMBER                  #IMPLIED
                    diglyph     NMTOKENS                #IMPLIED>
<!ELEMENT     def              - -     (#PCDATA | paratext | table)+>

<!ATTLIST      def            security         (uc | c | s |   ts)
#IMPLIED
                    restrict    NMTOKENS                #IMPLIED
                    release     NMTOKENS                #IMPLIED
                    codeword    NMTOKENS                #IMPLIED
                    scilevel    %yesorno;          "0"
                    diglyph     NMTOKENS                #IMPLIED>
<!ELEMENT     paratext         - -     (#PCDATA | %text; | (%list;))+>

<!ATTLIST      paratext       security         (uc | c | s |   ts)
#IMPLIED
                    config      NUTOKENS                #IMPLIED
                    restrict    NMTOKENS                #IMPLIED
                    release     NMTOKENS                #IMPLIED
                    codeword    NMTOKENS                #IMPLIED
                    scilevel    %yesorno;          "0"
                    diglyph     NMTOKENS                #IMPLIED
                    texttype    NUMBER                  #IMPLIED>
<!ELEMENT      para                - -       ((%spcpara;), (#PCDATA |
(%text;))+)>

<!ATTLIST     para            id         ID                  #IMPLIED
                    inschlvl    NUTOKEN                 #IMPLIED
                    delchlvl    NUTOKEN                 #IMPLIED
                    label       NMTOKEN                 #IMPLIED
                    texttype    NUMBER                  #IMPLIED
                    itemid      NMTOKEN                 #IMPLIED
                    config      NUTOKENS                #IMPLIED
                    skilltrk    CDATA                   #IMPLIED
                    hcp         %yesorno;          "0"
                    xref        IDREF                   #IMPLIED
                    security    (uc | c | s |  ts)       #IMPLIED
                    restrict    NMTOKENS                #IMPLIED
                    release     NMTOKENS                    #IMPLIED
                    codeword    NMTOKENS                    #IMPLIED
                    scilevel    %yesorno;          "0"
                    diglyph     NMTOKENS                    #IMPLIED>
<!ELEMENT     table           - -       (title, (tabdef |
                          stdtable), thead?,
                          tbody)>
<!ATTLIST     table       id               ID                    #IMPLIED
                    inschlvl    NUTOKEN                     #IMPLIED
                    delchlvl    NUTOKEN                     #IMPLIED
                    label       NMTOKEN                     #IMPLIED
                    texttype    NUMBER                      #IMPLIED
                    itemid      NMTOKEN                     #IMPLIED
                    config      NUTOKENS                    #IMPLIED
                    skilltrk    CDATA                       #IMPLIED
                    hcp         %yesorno;          "0"
                    xref        IDREF                       #IMPLIED
                    security    (uc | c | s |  ts)           #IMPLIED
                    restrict    NMTOKENS                    #IMPLIED
                    release     NMTOKENS                    #IMPLIED
                    codeword    NMTOKENS                    #IMPLIED
                    scilevel    %yesorno;          "0"
                    diglyph     NMTOKENS                    #IMPLIED>
<!ELEMENT     stdtable        - o     EMPTY>
<!ATTLIST     stdtable    name             ENTITY
#REQUIRED
                    security    (uc | c | s |  ts)           #IMPLIED
                    restrict    NMTOKENS                    #IMPLIED
                    release     NMTOKENS                    #IMPLIED
                    codeword    NMTOKENS                    #IMPLIED
                    scilevel    %yesorno;          "0"
                    diglyph     NMTOKENS                    #IMPLIED>
<!ELEMENT     tabdef          - -       (colbddef)+>
<!ATTLIST     tabdef      siderule        %yesorno;          "0"
                    colsep      %yesorno;          "0"
                    rowsep      %yesorno;          "0"
                    orient      (port | land)     "port"
                    width       (page | column)  "page"
                    security    (uc | c | s |  ts)           #IMPLIED
                    restrict    NMTOKFNS                    #IMPLIED
                    release     NMTOKENS                    #IMPLIED
                    codeword    NMTOKENS                    #IMPLIED
                    scilevel    %yesorno;          "0"
                    diglyph     NMTOKENS                    #IMPLIED>
<!ELEMENT     colbddef        - o     EMPTY>
<!ATTLIST     colbddef    colnum          NUMBER
#REQUIRED
                    datatype    CDATA                       #IMPLIED
                    left        %yesorno;          "0"
                    right       %yesorno;          "0"
                    center      %yesorno;          "0"
                    char        CDATA                       #IMPLIED
                    width       NUMBER                      #IMPLIED
                    leader      CDATA                       #IMPLIED
```

```
                colsep      %yesorno;              #IMPLIED
                rowhead     %yesorno;        "0"      >
<!ELEMENT   thead            - -      (row+)>
<!ATTLIST    thead          security        (uc | c | s |   ts)
#IMPLIED
                restrict    NMTOKENS                 #IMPLIED
                release     NMTOKENS                 #IMPLIED
                codeword    NMTOKENS                 #IMPLIED
                scilevel    %yesorno;        "0"
                diglyph     NMTOKENS                 #IMPLIED>
<!ELEMENT   tbody            - -      (row+)>
<!ATTLIST    tbody          security        (uc | c | s |   ts)
#IMPLIED
                restrict    NMTOKENS                 #IMPLIED
                release     NMTOKENS                 #IMPLIED
                codeword    NMTOKENS                 #IMPLIED
                scilevel    %yesorno;        "0"
                diglyph     NMTOKENS                 #IMPLIED>
<!ELEMENT   row              - -      (entry+)>
<!ATTLIST   row          number         NUMBER
#REQUIRED
                rowsep      %yesorno;              #IMPLIED
                security    (uc | c | s |   ts)      #IMPLIED
                restrict    NMTOKENS                 #IMPLIED
                release     NMTOKENS                 #IMPLIED
                codeword    NMTOKENS                 #IMPLIED
                scilevel    %yesorno;        "0"
                diglyph     NMTOKENS                 #IMPLIED>
<!ELEMENT   entry            - -      (#PCDATA) +(%list; | warning |

                                                     caution | note)>

<!ATTLIST   entry        col            NUTOKEN
#REQUIRED
                row         NMTOKEN                  #CURRENT
                rotate      NUMBER           "0"
                security    (uc | c | s |   ts)      #IMPLIED
                restrict    NMTOKENS                 #IMPLIED
                release     NMTOKENS                 #IMPLIED
                codeword    NMTOKENS                 #IMPLIED
                scilevel    %yesorno;        "0"
                diglyph     NMTOKENS                 #IMPLIED>
<!ELEMENT   causedby         - -      (#PCDATA, item+)>
<!ELEMENT   figure           - -      (graphic+, title,
                         legend?)>
<!ATTLIST   figure       id             ID                 #IMPLIED
                inschlvl    NUTOKEN                  #IMPLIED
                delchlvl    NUTOKEN                  #IMPLIED
                label       NMTOKEN                  #IMPLIED
                texttype    NUMBER                   #IMPLIED
                itemid      NMTOKEN                  #IMPLIED
                config      NUTOKENS                 #IMPLIED
                skilltrk    CDATA                    #IMPLIED
                hcp         %yesorno;        "0"
```

```
                  xref        IDREF                   #IMPLIED
                  security    (uc | c | s |  ts)       #IMPLIED
                  restrict    NMTOKENS                #IMPLIED
                  release     NMTOKENS                #IMPLIED
                  codeword    NMTOKENS                #IMPLIED
                  scilevel    %yesorno;          "0"
                  diglyph     NMTOKENS                #IMPLIED>
<!ELEMENT    legend          - -      (deflist)>
<!ATTLIST     legend        security         (uc | c | s |  ts)
#IMPLIED
                  restrict    NMTOKENS                #IMPLIED
                  release     NMTOKENS                #IMPLIED
                  codeword    NMTOKENS                #IMPLIED
                  scilevel    %yesorno;          "0"
                  diglyph     NMTOKENS                #IMPLIED>
<!ELEMENT    body            - -      (section+)>
<!ATTL1ST     body          security         (uc | c | s |  ts)
#IMPLIED
                  restrict    NMTOKENS                #IMPLIED
                  release     NMTOKENS                #IMPLIED
                  codeword    NMTOKENS                #IMPLIED
                  scilevel    %yesorno;          "0"
                  diglyph     NMTOKENS                #IMPLIED>
<!ELEMENT    section         - -      (title, (title?,
                          (paratext | (%spcpara;)
                          | %list;)+)+, fltdesc*,
                          fltiso*, (figure,
                          fltproc)*)>
<!ATTLIST    section     id            ID                 #IMPLIED
                  inschlvl    NUTOKEN                 #IMPLIED
                  delchlvl    NUTOKEN                 #IMPLIED
                  label       NMTOKEN                 #IMPLIED
                  texttype    NUMBER                  #IMPLIED
                  itemid      NMTOKEN                 #IMPLIED
                  config      NUTOKENS                #IMPLIED
                  skilltrk    CDATA                   #IMPLIED
                  hcp         %yesorno;          "0"
                  xref        IDREF                   #IMPLIED
                  security    (uc | c | s |  ts)       #IMPLIED
                  restrict    NMTOKENS                #IMPLIED
                  release     NMTOKENS                #IMPLIED
                  codeword    NMTOKENS                #IMPLIED
                  scilevel    %yesorno;          "0"
                  diglyph     NMTOKENS                #IMPLIED>
<!ELEMENT    fltdesc         - -      (title, descunit+)>
<!ATTLIST    fltdesc     fltcode        CDATA
#REQUIRED
                  security    (uc | c | s |  ts)       #IMPLIED
                  restrict    NMTOKENS                #IMPLIED
                  release     NMTOKENS                #IMPLIED
                  codeword    NMTOKENS                #IMPLIED
                  scilevel    %yesorno;          "0"
                  diglyph     NMTOKENS                #IMPLIED>
<!ELEMENT    descunit        - -      ((#PCDATA),descunit*)>
```

```
<!ATTLIST    descunit    add2flt       CDATA
#IMPLIED
                  security    (uc | c | s |  ts)     #IMPLIED
                  restrict    NMTOKENS              #IMPLIED
                  release     NMTOKENS              #IMPLIED
                  codeword    NMTOKENS              #IMPLIED
                  scilevel    %yesorno;        "0"
                  diglyph     NMTOKENS              #IMPLIED>
<!ELEMENT    fltiso          - -      (sppteqpt?, (%spcpara;),
                         causedby, (taskinfo,
                         ((test, respgrp+) | table))+)>
<!ATTLIST    fltiso      fltcode       NUTOKEN
#REQUIRED
                  security    (uc | c | s |  ts)     #IMPLIED
                  restrict    NMTOKENS              #IMPLIED
                  release     NMTOKENS              #IMPLIED
                  codeword    NMTOKENS              #IMPLIED
                  scilevel    %yesorno;        "0"
                  diglyph     NMTOKENS              #IMPLIED>
<!ELEMENT    fltproc         - -      (sppteqpt?, (%spcpara;),
                         causedby?, (taskinfo,
                         (test, respgrp+))+)>
<!ATTLIST    fltproc     fltcode       NUTOKEN
#REQUIRED
                  security    (uc | c | s |  ts)     #IMPLIED
                  restrict    NMTOKENS              #IMPLIED
                  release     NMTOKENS              #IMPLIED
                  codeword    NMTOKENS              #IMPLIED
                  scilevel    %yesorno;        "0"
                  diglyph     NMTOKENS              #IMPLIED>
<!ELEMENT    sppteqpt        - -      (paratext)>
<!ELEMENT    taskinfo        - -      (title?, %spcpara;,
                       (stepgrp | step1)* )>
<!ATTLIST    taskinfo    tasktype      CDATA
#IMPLIED
                  id          ID               #IMPLIED
                  inschlvl    NUTOKEN               #IMPLIED
                  delchlvl    NUTOKEN               #IMPLIED
                  label       NMTOKEN               #IMPLIED
                  texttype    NUMBER                #IMPLIED
                  itemid      NMTOKEN               #IMPLIED
                  config      NUTOKENS              #IMPLIED
                  skilltrk    CDATA                 #IMPLIED
                  hcp         %yesorno;        "0"
                  xref        IDREF                 #IMPLIED
                  security    (uc | c | s |  ts)     #IMPLIED
                  restrict    NMTOKENS              #IMPLIED
                  release     NMTOKENS              #IMPLIED
                  codeword    NMTOKENS              #IMPLIED
                  scilevel    %yesorno;        "0"
                  diglyph     NMTOKENS              #IMPLIED>
<!ELEMENT    stepgrp         - -      (step1, step1+)>
<!ATTLIST    stepgrp     id            ID                   #IMPLIED
                  inschlvl    NUTOKEN               #IMPLIED
```

```
                     delchlvl     NUTOKEN                         #IMPLIED
                     label        NMTOKEN                         #IMPLIED
                     texttype     NUMBER                          #IMPLIED
                     itemid       NMTOKEN                         #IMPLIED
                     config       NUTOKENS                        #IMPLIED
                     skilltrk     CDATA                           #IMPLIED
                     hcp          %yesorno;             "0"
                     xref         IDREF                           #IMPLIED
                     security     (uc | c | s |  ts)               #IMPLIED
                     restrict     NMTOKENS                        #IMPLIED
                     release      NMTOKENS                        #IMPLIED
                     codeword     NMTOKENS                        #IMPLIED
                     scilevel     %yesorno;             "0"
                     diglyph      NMTOKENS                        #IMPLIED>
<!ELEMENT      step1              - -      ((%spcpara;),paratext, step2*)>

<!ATTLIST      step1      id                    ID                    #IMPLIED
                     inschlvl     NUTOKEN                         #IMPLIED
                     delchlvl     NUTOKEN                         #IMPLIED
                     label        NMTOKEN                         #IMPLIED
                     itemid       NMTOKEN                         #IMPLIED
                     config       NUTOKENS                        #IMPLIED
                     skilltrk     CDATA                           #IMPLIED
                     hcp          %yesorno;             "0"
                     xref         IDREF                           #IMPLIED
                     security     (uc | c | s |  ts)               #IMPLIED
                     restrict     NMTOKENS                        #IMPLIED
                     release      NMTOKENS                        #IMPLIED
                     codeword     NMTOKENS                        #IMPLIED
                     scilevel     %yesorno;             "0"
                     diglyph      NMTOKENS                        #IMPLIED
                     ipi          NAME                            #IMPLIED
                     person       NAME                            #IMPLIED
                     texttype     NUMBER                          #IMPLIED>
<!ELEMENT      step2              - -      ((%spcpara;), paratext, step3*)>

<!ATTLIST      step2      id                    ID                    #IMPLIED
                     inschlvl     NUTOKEN                         #IMPLIED
                     delchlvl     NUTOKEN                         #IMPLIED
                     label        NMTOKEN                         #IMPLIED
                     itemid       NMTOKEN                         #IMPLIED
                     config       NUTOKENS                        #IMPLIED
                     skilltrk     CDATA                           #IMPLIED
                     hcp          %yesorno;             "0"
                     xref         IDREF                           #IMPLIED
                     security     (uc | c | s |  ts)               #IMPLIED
                     restrict     NMTOKENS                        #IMPLIED
                     release      NMTOKENS                        #IMPLIED
                     codeword     NMTOKENS                        #IMPLIED
                     scilevel     %yesorno;             "0"
                     diglyph      NMTOKENS                        #IMPLIED
                     ipi          NAME                            #IMPLIED
                     person       NAME                            #IMPLIED
                     texttype     NUMBER                          #IMPLIED>
```

```
<!ELEMENT    step3          - -     ((%spcpara;), paratext)>
<!ATTLIST    step3       id             ID                #IMPLIED
                  inschlvl   NUTOKEN                  #IMPLIED
                  delchlvl   NUTOKEN                  #IMPLIED
                  label      NMTOKEN                  #IMPLIED
                  itemid     NMTOKEN                  #IMPLIED
                  config     NUTOKENS                 #IMPLIED
                  skilltrk   CDATA                    #IMPLIED
                  hcp        %yesorno;          "0"
                  xref       IDREF                    #IMPLIED
                  security   (uc | c | s |   ts)       #IMPLIED
                  restrict   NMTOKENS                 #IMPLIED
                  release    NMTOKENS                 #IMPLIED
                  codeword   NMTOKENS                 #IMPLIED
                  scilevel   %yesorno;          "0"
                  diglyph    NMTOKENS                 #IMPLIED
                  ipi        NAME                     #IMPLIED
                  person     NAME                     #IMPLIED
                  texttype   NUMBER                   #IMPLIED>
<!ELEMENT    test           - -     (#PCDATA, response+)>
<!ELEMENT    response       - o     EMPTY>
<!ATTLIST    response    rsvpval     CDATA                        "Yes"

               rsvpref    IDREF              #REQUIRED
               xref       IDREF                   #IMPLIED>

<!ELEMENT    respgrp          - -     (taskinfo, (test | table)?)>

<!ATTLIST    respgrp     id       ID                      #REQUIRED>


<!ELEMENT    figindex       - -     (#PCDATA)>
<!ATTLIST     figindex      security        (uc | c | s |   ts)
#IMPLIED
                  restrict   NMTOKENS                 #IMPLIED
                  release    NMTOKENS                 #IMPLIED
                  codeword   NMTOKENS                 #IMPLIED
                  scilevel   %yesorno;          "0"
                  diglyph    NMTOKENS                 #IMPLIED>
<!ELEMENT    docno          - -     (#PCDATA)>
<!ATTLIST     docno         security        (uc | c | s |   ts)
#IMPLIED
                  restrict   NMTOKENS                 #IMPLIED
                  release    NMTOKENS                 #IMPLIED
                  codeword   NMTOKENS                 #IMPLIED
                  scilevel   %yesorno;          "0"
                  diglyph    NMTOKENS                 #IMPLIED>
<!ELEMENT    sssn           - -     (#PCDATA)>
<!ATTLIST     sssn          security        (uc | c | s |   ts)
#IMPLIED
                  restrict   NMTOKENS                 #IMPLIED
                  release    NMTOKENS                 #IMPLIED
                  codeword   NMTOKENS                 #IMPLIED
                  scilevel   %yesorno;          "0"
```

```
                        diglyph     NMTOKENS                  #IMPLIED>
<!ELEMENT     tool                - -      (#PCDATA)>
<!ATTLIST      tool              security          (uc | c | s |    ts)
#IMPLIED
                        restrict    NMTOKENS                  #IMPLIED
                        release     NMTOKENS                  #IMPLIED
                        codeword    NMTOKENS                  #IMPLIED
                        scilevel    %yesorno;          "0"
                        diglyph     NMTOKENS                  #IMPLIED>
<!ELEMENT     testeq              - -      (#PCDATA)>
<!ATTLIST      testeq            security          (uc | c | s |    ts)
#IMPLIED
                        restrict    NMTOKENS                  #IMPLIED
                        release     NMTOKENS                  #IMPLIED
                        codeword    NMTOKENS                  #IMPLIED
                        scilevel    %yesorno;          "0"
                        diglyph     NMTOKENS                  #IMPLIED>
<!ELEMENT     material            - -      (#PCDATA)>
<!ATTLIST      material          security          (uc | c | s |    ts)
#IMPLIED
                        restrict    NMTOKENS                  #IMPLIED
                        release     NMTOKENS                  #IMPLIED
                        codeword    NMTOKENS                  #IMPLIED
                        scilevel    %yesorno;          "0"
                        diglyph     NMTOKENS                  #IMPLIED>
<!ELEMENT     torquevl            - -      (#PCDATA)>
<!ATTLIST      torquevl          security          (uc | c | s |    ts)
#IMPLIED
                        restrict    NMTOKENS                  #IMPLIED
                        release     NMTOKENS                  #IMPLIED
                        codeword    NMTOKENS                  #IMPLIED
                        scilevel    %yesorno;          "0"
                        diglyph     NMTOKENS                  #IMPLIED>
<!ELEMENT     stchg             - o     EMPTY>
<!ATTLIST     stchg          level          NUMBER
#IMPLIED
                        status      (add | delete)             #IMPLIED
                        security    (uc | c | s |    ts)       #IMPLIED
                        restrict    NMTOKENS                  #IMPLIED
                        release     NMTOKENS                  #IMPLIED
                        codeword    NMTOKENS                  #IMPLIED
                        scilevel    %yesorno;          "0"
                        diglyph     NMTOKENS                  #IMPLIED>
<!ELEMENT     endchg            - o     EMPTY>
<!ATTLIST     endchg         level          NUMBER
#IMPLIED
                        status      (add | delete)             #IMPLIED
                        security    (uc | c | s |    ts)       #IMPLIED
                        restrict    NMTOKENS                  #IMPLIED
                        release     NMTOKENS                  #IMPLIED
                        codeword    NMTOKENS                  #IMPLIED
                        scilevel    %yesorno;          "0"
                        diglyph     NMTOKENS                  #IMPLIED>
<!ELEMENT     subscrpt            - -      (#PCDATA, subscrpt?,
                                supscrpt?)>
<!ATTLIST     subscrpt     security        (uc | c | s |   ts)
#IMPLIED
                  restrict    NMTOKENS                 #IMPLIED
                  release     NMTOKENS                 #IMPLIED
                  codeword    NMTOKENS                 #IMPLIED
                  scilevel    %yesorno;         "0"
                  diglyph     NMTOKENS                 #IMPLIED>
<!ELEMENT     supscrpt          - -     (#PCDATA, subscrpt?,
                                supscrpt?)>
<!ATTLIST     supscrpt     security        (uc | c | s |   ts)
#IMPLIED
                  restrict    NMTOKENS                 #IMPLIED
                  release     NMTOKENS                 #IMPLIED
                  codeword    NMTOKENS                 #IMPLIED
                  scilevel    %yesorno;         "0"
                  diglyph     NMTOKENS                 #IMPLIED>
<!ELEMENT     config            - o     EMPTY>
<!ATTLIST     config       code            CDATA            #REQUIRED

                  security    (uc | c | s |  ts)       #IMPLIED
                  restrict    NMTOKENS                 #IMPLIED
                  release     NMTOKENS                 #IMPLIED
                  codeword    NMTOKENS                 #IMPLIED
                  scilevel    %yesorno;         "0"
                  diglyph     NMTOKENS                 #IMPLIED>
<!ELEMENT     xref              - o     EMPTY>
<!ATTLIST     xref         xref            CDATA
#REQUIRED
                  security    (uc | c | s |  ts)       #IMPLIED
                  restrict    NMTOKENS                 #IMPLIED
                  release     NMTOKENS                 #IMPLIED
                  codeword    NMTOKENS                 #IMPLIED
                  scilevel    %yesorno;         "0"
                  diglyph     NMTOKENS                 #IMPLIED>
<!ELEMENT     graphic           - o     EMPTY>
<!ATTLIST     graphic      boardno         CDATA
#REQUIRED
                  width       NUTOKEN                  #IMPLIED
                  depth       NUTOKEN                  #IMPLIED
                  window      CDATA                    #IMPLIED
                  hscale      NUTOKEN                  #IMPLIED
                  vscale      NUTOKEN                  #IMPLIED
                  text        CDATA                    #IMPLIED
                  rotation    NUMBER            "0"
                  security    (uc | c | s |  ts)       #IMPLIED
                  restrict    NMTOKENS                 #IMPLIED
                  release     NMTOKENS                 #IMPLIED
                  codeword    NMTOKENS                 #IMPLIED
                  scilevel    %yesorno;         "0"
                  diglyph     NMTOKENS                 #IMPLIED>
<!ELEMENT     partno            - -     (#PCDATA)>
<!ATTLIST     partno       security        (uc | c | s |   ts)
#IMPLIED
```

```
                 restrict   NMTOKENS                #IMPLIED
                 release    NMTOKENS                #IMPLIED
                 codeword   NMTOKENS                #IMPLIED
                 scilevel   %yesorno;          "0"
                 diglyph    NMTOKENS                #IMPLIED>
<!ELEMENT   modelno       - -     (#PCDATA)>
<!ATTLIST    modelno      security        (uc | c | s |   ts)
#IMPLIED
                 restrict   NMTOKENS                #IMPLIED
                 release    NMTOKENS                #IMPLIED
                 codeword   NMTOKENS                #IMPLIED
                 scilevel   %yesorno;          "0"
                 diglyph    NMTOKENS                #IMPLIED>
<!ELEMENT   warning       - -     (graphic?, (paratext))>
<!ATTLIST   warning     name          ENTITY                  #CONREF

                 type       NAME                    #IMPLIED
                 xref       IDREF                   #IMPLIED
                 vital      (y | n)            "n"
                 security   (uc | c | s |  ts)       #IMPLIED
                 restrict   NMTOKENS                #IMPLIED
                 release    NMTOKENS                #IMPLIED
                 codeword   NMTOKENS                #IMPLIED
                 scilevel   %yesorno;          "0"
                 diglyph    NMTOKENS                #IMPLIED>
<!ELEMENT   caution       - -     (graphic?, (paratext))>
<!ATTLIST   caution     name          ENTITY                  #CONREF

                 type       NAME                    #IMPLIED
                 xref       IDREF                   #IMPLIED
                 security   (uc | c | s |  ts)       #IMPLIED
                 restrict   NMTOKENS                #IMPLIED
                 release    NMTOKENS                #IMPLIED
                 codeword   NMTOKENS                #IMPLIED
                 scilevel   %yesorno;          "0"
                 diglyph    NMTOKENS                #IMPLIED>
<!ELEMENT   note          - -     (graphic?, (paratext))>
<!ATTLIST   note        name          ENTITY                  #CONREF

                 type       NAME                    #IMPLIED
                 xref       IDREF                   #IMPLIED
                 security   (uc | c | s |  ts)       #IMPLIED
                 restrict   NMTOKENS                #IMPLIED
                 release    NMTOKENS                #IMPLIED
                 codeword   NMTOKENS                #IMPLIED
                 scilevel   %yesorno;          "0"
                 diglyph    NMTOKENS                #IMPLIED>
<!ELEMENT   stemg         - o    EMPTY>
<!ELEMENT   endemg        - o    EMPTY>
<!ELEMENT   stemph        - o    EMPTY>
<!ATTLIST   stemph      type          NUMBER
#REQUIRED   >
<!ELEMENT   endemph       - o    EMPTY>
<!ATTLIST   endemph     type          NUMBER
```

```
#REQUIRED    >
 ]>
```

APPENDIX D

INTERMEDIATE MAINTENANCE INSTRUCTIONS DOCUMENT TYPE DEFINITION

```
<!DOCTYPE tmimi [

<!ENTITY % yesorno "NUMBER">

<!ENTITY % bodyatt  "id          ID              #IMPLIED
               inschlvl  NUTOKEN         #IMPLIED
               delchlvl  NUTOKEN         #IMPLIED
               label          NMTOKEN         #IMPLIED
               texttype  NUMBER          #IMPLIED
               itemid         NMTOKEN         #IMPLIED
               config         NUTOKENS   #IMPLIED
               skilltrk  CDATA           #IMPLIED
               hcp       %yesorno; '0'
               xref      IDREF           #IMPLIED" >


<!ENTITY % colfmt   "colnum         NUMBER          #REQUIRED
               datatype  CDATA           #IMPLIED
               left      %yesorno; '0'
               right          %yesorno; '0'
               center         %yesorno; '0'
               char      CDATA           #IMPLIED
               width          NUMBER          #IMPLIED
               leader         CDATA           #IMPLIED
               colsep         %yesorno; #IMPLIED
               rowhead        %yesorno; '0' "          >

<!ENTITY % contdef  "TBD" >

<!ENTITY % def           "" >


<!ENTITY % list          "seqlist | randlist | deflist"
     >


<!ENTITY % secur    "security (uc | c |
                          s | ts)  #IMPLIED
               restrict  NMTOKENS  #IMPLIED
               release        NMTOKENS  #IMPLIED
               codeword  NMTOKENS  #IMPLIED
               scilevel  %yesorno  '0'
               diglyph        NMTOKENS  #IMPLIED" >

<!ENTITY % spcpara  "(warning*, caution*, note*)"          >

<!ENTITY % change   "stchg | endchg"                    >
<!ENTITY % asyntxt  "stemg | endemg | %change; |
               stemph | endemph | config"                    >
<!ENTITY % nums           "partno | modelno | sssn | figindex"
     >
<!ENTITY % text     "%asyntxt; |
               %nums; | tool | testeq |
               material | torquevl |
               xref | graphic |
               subscrpt | supscrpt"                          >

<                    !                    -                    -
*****************************************************************
     -->
<!--           End Entities                          -->
<                    !                    -                    -
*****************************************************************
     -->



<                    !                    -                    -
*****************************************************************
     -->
<!-- Definitions of Elements and their Attributes           -->
<                    !                    -                    -
*****************************************************************
     -->

<!ELEMENT tmimi          - -      (front, body)              >
<!--ELENOTE    tmimi        Technical Manual : Intermidiate
                    Maintenance Instructions      -->

<!ATTLIST tmimi          status         (revision |
                         change |
                         prelim |
                         draft |
                         formal)        "formal"
                     revno          NUMBER         "0"
                     chgno          NUMBER         "0"
                       %secur;                               >
<!--ATTNOTE tmimi   status         "This is the status of the
                         document at the time of
                         composition, and is usually
                         reflected on the title page."
               revno          "Revision number of the doc."
               chgno          "Change level of the document."
               %secur;        "defined in entities          -->
```

```
<!ELEMENT front             - -       (idinfo,
                            lep,
                            verstat?,
                            contents,
                            loi, lot,
                            intro?)               >

<!ATTLIST front             %secur;                          >


<!ELEMENT idinfo       - -       (pubno+,
                       prepubno?,
                       %spcpara;,
                       titleblk,
                       (mfr, contrno)+,
                       docmfr?,
                       notice+,
                       pubdate,
                       chgdate?)            >

<!ATTLIST idinfo       %secur;                          >


<!ELEMENT pubno             - -       (user, docno)          >

<!ATTLIST pubno             %secur;                          >


<!ELEMENT prepubno     - -       (user, docno)          >

<!ATTLIST prepubno     %secur;                          >


<!ELEMENT user         - -       (#PCDATA)         >

<!ATTLIST user         %secur;                          >


<!ELEMENT titleblk     - -       (doctype,
                       maintlvl*,
                       prtitle)             >

<!ATTLIST titleblk     %secur;                          >
```

```
<!ELEMENT doctype   - -       (#PCDATA) +(%text;) >
<!ATTLIST doctype   %secur;                            >

<!ELEMENT maintlvl  - -       (#PCDATA) +(%text;)           >
<!ATTLIST maintlvl  %secur;                            >

<!ELEMENT sssnrng   - -       (#PCDATA) +(%text;)           >
<!ATTLIST sssnrng   %secur;                            >

<!ELEMENT prtitle   - -       (nomen, type?)           >
<!ATTLIST prtitle   %secur;                            >

<!ELEMENT nomen          - -       (#PCDATA)           >
<!ATTLIST nomen          %secur;                            >
<!ELEMENT typedes   - -       (#PCDATA)           >
<!ATTLIST typedes   %secur;                            >
<!ELEMENT altdes    - -       (#PCDATA)           >
<!ATTLIST altdes    %secur;                            >
<!ELEMENT use       - -       (#PCDATA)           >
<!ATTLIST use       %secur;                            >
<!ELEMENT specs          - -       (#PCDATA)           >
<!ATTLIST specs          %secur;                            >
<!ELEMENT type      - -       (#PCDATA)           >
```

```
<!ATTLIST type      %secur;                                   >
<!ELEMENT charac    - -        (para+)                        >
<!ATTLIST charac    %secur;                                   >
<!ELEMENT geninfo   - -        (para+)                        >
<!ATTLIST geninfo   %secur;                                   >
<!ELEMENT purpose   - -        (para+)                        >
<!ATTLIST purpose   %secur;                                   >

<!ELEMENT factdata  - -        (para+)                        >
<!ATTLIST factdata  %secur;                                   >


<!ELEMENT mfr       - -        (#PCDATA)                >
<!ATTLIST mfr       %secur;                                   >


<!ELEMENT docmfr    - -        (#PCDATA)                >
<!ATTLIST docmfr    %secur;                                   >


<!ELEMENT contrno   - -        (#PCDATA)                >
<!ATTLIST contrno   %secur;                                   >


<!ELEMENT notice    - -        ((#PCDATA) |
                          (%list;))*     +(%text;) >

<!ATTLIST notice    type       (dist      |
                          auth |
                          offic      |
                          branch     |
                          pgclass    |
                         discl     |
                         supersed|
                         effdate   |
                         suppl     |
                         nopg |
                         noclaspg) #IMPLIED
               %secur;                            >


<!ELEMENT pubdate   - -       (#PCDATA)           >

<!ATTLIST pubdate   %secur;                            >


<!ELEMENT chgdate   - -       (#PCDATA)           >

<!ATTLIST chgdate   %secur;                            >


<!ELEMENT date      - -       (#PCDATA)           >

<!ATTLIST date      %secur;                            >


<!ELEMENT lep       - o       EMPTY                    >

<!ELEMENT loi       - o       EMPTY                    >

<!ELEMENT lot       - o       EMPTY                    >

<!ELEMENT verstat   - -       (para,
                         (para |
                         %list; |
                         %spcpara;)*)             >

<!ATTLIST verstat   %secur;                            >


<!ELEMENT contents  - o       EMPTY                    >
```

```
<!ELEMENT intro            - -      (title,
                          (title?,
                          (para |
                          abbrsect |
                          symsect |
                          tctorec |
                          (%list;) )+ )+ )          >

<!ATTLIST intro           %secur;                           >


<!ELEMENT seqlist   - -      (title?, item*)                >

<!ATTLIST seqlist   label        CDATA          #IMPLIED
               %secur;                          >


<!ELEMENT randlist  - -      (title?, item*)                >

<!ATTLIST randlist  %secur;                          >


<!ELEMENT deflist   - -      (term, def)+                >

<!ATTLIST deflist   %secur;                          >


<!ELEMENT item        - -    (#PCDATA) +(%text;)            >
<!ATTLIST item        %bodyatt;
               %secur;                          >


<!ELEMENT symsect   - -      (deflist)          >

<!ATTLIST symsect   %secur;                          >


<!ELEMENT abbrsect  - -      (deflist)          >

<!ATTLIST abbrsect  %secur;                          >
```

```
<!ELEMENT term      - -       (#PCDATA) +(%text;) >

<!ATTLIST term      %secur;                            >


<!ELEMENT tctorec   - -       (pubno,
                         title,
                         date?)+                  >

<!ATTLIST tctorec   %secur;                            >


<!ELEMENT title          - -       (#PCDATA) +(%text;)          >

<!ATTLIST title          %secur;
               texttype  NUMBER         #IMPLIED  >


<!ELEMENT def       - -       ((#PCDATA) |
                         table)+        -(%list;)
                              +(%text;) >

<!ATTLIST def       %secur;                            >


<!ELEMENT para      - -       (title?,
                         (#PCDATA),
                         (%list; |
                         %spcpara;)*)+      +(%text;) >

<!ATTLIST para      %bodyatt;
               %secur;                            >


<!ELEMENT chart          - -       ( title,
                         ( tabdef |
                         stdtable ),
                         thead?,
                         tbody )            >

<!ATTLIST chart          %bodyatt;
```

```
                    %secur;                                   >


<!ELEMENT table              - -        (title,
                             (tabdef |
                             stdtable),
                             thead?,
                             tbody)            +(warning |
                                        caution |
                                        note)                 >

<!ATTLIST table              %bodyatt;
                    %secur;                                   >


<!ELEMENT stdtable    - o        EMPTY                            >

<!ATTLIST stdtable    name       ENTITY          #REQUIRED
                    %secur;                                   >


<!ELEMENT tabdef      - -        (colbddef)*                      >

<!ATTLIST tabdef      siderule   %yesorno; "0"
                    colsep       %yesorno; "0"
                    rowsep       %yesorno; "0"
                    orient       (port | land)  "port"
                    width        (page | column) "page"
                    %secur;                                   >


<!ELEMENT colbddef    - o        EMPTY                            >

<!ATTLIST colbddef    %colfmt;                                >


<!ELEMENT thead              - -        (row+)                        >

<!ATTLIST thead              %secur;                                  >


<!ELEMENT tbody              - -        (row+)                        >
```

```
<!ATTLIST tbody          %secur;                                      >


<!ELEMENT row        - -        (entry+)              >

<!ATTLIST row        number          NUMBER           #REQUIRED
             rowsep          %yesorno; #IMPLIED
             %secur;                                  >


<!ELEMENT entry              - -       (#PCDATA) -(figure |
                                       table |
                                       chart)
                                       +(%text;) >

<!ATTLIST entry              col       NUTOKEN          #REQUIRED
             row             NMTOKEN          #CURRENT
             rotate               NUMBER           "0"
             %secur;                                  >


<!ELEMENT figure     - -         (graphic+,
                         title,
                         legend?)             >

<!ATTLIST figure     %bodyatt;
             %secur;                                  >


<!ELEMENT legend     - -        (deflist)             >

<!ATTLIST legend     %secur;                                  >


<!ELEMENT body       - -        (descsect,
                         toolsect,
                         theory?,
                         tiein?,
                         prepsect?,
                         chkoutsect?,
                         section+,
                         ipb)                         >
```

```
<!-- The body of an intermediate maintenance instruction set
     should consist of a Description and a Special tool section
     optionally followed by Theory, tiein, preparation, and check

     out.  These optinal sections will be followed by one or
     more setions of maintenance instructions, and an Illustrated

     Parts Breakdown.                                   -->

<!ATTLIST body      %secur;                                 >

<!ELEMENT descsect   - -      (geninfo, purpose,
                          factdata, charac)  +(figure |
                                         chart |
                                         table)          >
<!--ELENOTE descsect         This  section  gives  an  overall
description
                    of the system                       -->
<!ATTLIST descsect  %secur;                                 >

<!ELEMENT toolsect   - -      (geninfo,
                          spectool,
                          speceqpt*,
                          imprtool?,
                          consume?)          +(figure |
                                         chart |
                                         table )   >

<!--ELENOTE toolsect          This section is for special tools,

                    test equipment, and consumables          -->
<!ATTLIST toolsect  %secur;                                 >

<!ELEMENT imprtool   - -      (para*, toollist)         >
<!ATTLIST imprtool  %secur;                                 >

<!ELEMENT spectool   - -      (para*, toollist)         >
<!ATTLIST spectool  %secur;                                 >

<!ELEMENT toollist   - -      (partno,
                          figindex,
                          nomen,
                          use)+            >
```

```
<!--ELENOTE toollist          This is a tabular form list with the

                    following format:
                    SPECIAL TOOLS LIST
Part(Tool)          Figure &
Number              Index No.      Nomenclature        Use
                                             -->
<!ATTLIST toollist  %secur;                            >

<!ELEMENT speceqpt  - -       (para*, eqptlist)   +(figure |
                                     chart |
                                     table) >
<!ATTLIST speceqpt  %secur;                            >


<!ELEMENT eqptlist  - -       (typedes,
                          altdes,
                          figindex,
                          nomen,
                          use)+                >
<!--ELENOTE eqptlist          This is a tabular form list with the

                    following format:
                    TEST EQUIPMENT LIST
               Alternate
Type           Type      Figure &
Designation         Designation    Index No. Nomenclature   Use
                                             -->
<!ATTLIST eqptlist  %secur;                            >


<!ELEMENT consume   - -       (nomen,
                          specs,
                          partno)+     +(figure |
                                     chart |
                                     table )   >
<!--ELENOTE consume      This is a tabular form list in alphabetic

                    order, with the     following format:
Nomenclature             Specification(s)    Part number and FMSC

                                             -->
<!ATTLIST consume   %secur;                            >


<!ELEMENT theory    - -       (para+)        +(figure |
                                 chart |
                                   table)          >
<!--ELENOTE theory         This section describes the theory
                    of operation of the equipment and
                    accessories                    -->
<!ATTLIST theory    %bodyatt;
               %secur;                             >


<!ELEMENT tiein          - -       (para+)         +(figure |
                                   chart |
                                   table)          >
<!--ELENOTE tiein          This section describes the system
                    tiein of equipment and accessories -->
<!ATTLIST tiein          %bodyatt;
               %secur;                             >


<!ELEMENT prepsect  - -       (geninfo,
                          (para | taskinfo)+ )     +(figure |
                                        chart |
                                        table)     >
<!--ELENOTE prepsect          This    section    describes    and
illustrates
                    the procedures to uncrate and unpack
                    the equipment for intermidiate maint.
          -->
<!ATTLIST prepsect  %bodyatt;
               %secur;                             >


<!ELEMENT chkoutsect     - -       (geninfo,
                          (para | taskinfo |
                           subsect)+ )     +(figure |
                                        chart |
                                        table)               >
<!--ELENOTE chkoutsect        This section contains steps for
                    performance testing and locating
                    and identifying malfunctions        -->
<!ATTLIST chkoutsect     %bodyatt;
               %secur;                             >


<!ELEMENT section   - -       (title,
                         subsect+ )        +(figure |
                                        chart |
                                        table)     >
<!--ELENOTE section        "These are sections which describe
                    and procedurally detail tasks involved
                    in maintenance and repair          -->

<!ATTLIST section   %bodyatt;
               %secur;                             >

<!ELEMENT ipb       - o       EMPTY                     >
<!-- put in the IPB definition we came up with before
-->
<!--ELENOTE ipb          This section consists of a Illustrated
               Parts Breakdown in accordance with
               MIL-M-38807.                        -->


<!ELEMENT subsect   - -       (title,
                         para+,
                         taskinfo*,
                         para* )                   >
<!ATTLIST subsect   %bodyatt;
               %secur;                             >


<!ELEMENT taskinfo  - -       (title?,
                         (para|
                         step1)+)             >
<!ATTLIST taskinfo  %bodyatt;
               %secur;                             >


<!ELEMENT step1          - -       ((%spcpara;),
                         (#PCDATA),
                         step2*)   +(figure |
                                     %text;) >
<!ATTLIST step1          %bodyatt;
               %secur;
               ipi       NAME      #IMPLIED
               person         NAME      #IMPLIED   >


<!ELEMENT step2          - -       ((%spcpara;),
                         (#PCDATA),
                         (step3)*) +(%text;) >
<!ATTLIST step2          %bodyatt;
               %secur;
               ipi       NAME      #IMPLIED
               person         NAME      #IMPLIED  >


<!ELEMENT step3          - -       ((%spcpara;),
                         (#PCDATA))     +(%text;) >

<!ATTLIST step3          %bodyatt;
               %secur;
               ipi       NAME      #IMPLIED
               person         NAME      #IMPLIED  >



<!ELEMENT docno          - -       (#PCDATA)           >

<!ATTLIST docno          %secur;                            >



<!ELEMENT warning   - -       (graphic?,
                         (para*) )           >

<!ATTLIST warning   name      ENTITY         #CONREF
               type      NAME      #IMPLIED
               xref      IDREF          #IMPLIED
               vital          (y | n)        "n"
               %secur;                            >


<!ELEMENT caution   - -       (graphic?,
                         (para*) )           >
<!ATTLIST caution   name      ENTITY         #CONREF
               type      NAME      #IMPLIED
               xref      IDREF          #IMPLIED
               %secur;                            >


<!ELEMENT note      - -       (graphic?,
                         (para*) )           >
<!ATTLIST note      name      ENTITY         #CONREF
               type      NAME      #IMPLIED
               xref      IDREF          #IMPLIED
```

```
                 %secur;                                  >


<!ELEMENT stemg           - o       EMPTY                          >

<!ELEMENT endemg     - o       EMPTY                          >

<!ELEMENT stemph     - o       EMPTY                          >
<!ATTLIST stemph     type      NUMBER          #REQUIRED >

<!ELEMENT endemph    - o       EMPTY                          >
<!ATTLIST endemph    type      NUMBER          #REQUIRED >


<!ELEMENT config     - o       EMPTY                          >

<!ATTLIST config     code      CDATA           #REQUIRED
                 %secur;                                  >


<!ELEMENT xref       - o       EMPTY                          >
<!ATTLIST xref       xref      CDATA           #REQUIRED
                 %secur;                                  >


<!ELEMENT graphic    - o       EMPTY                          >
<!ATTLIST graphic    boardno        CDATA           #REQUIRED
                 width          NUTOKEN         #IMPLIED
                 depth          NUTOKEN         #IMPLIED
                 window         CDATA           #IMPLIED
                 hscale         NUTOKEN         #IMPLIED
                 vscale         NUTOKEN         #IMPLIED
                 text      CDATA           #IMPLIED
                 rotation  NUMBER          "0"
                 %secur;                                  >

<!ELEMENT tool       - -       (#PCDATA)                 >

<!ATTLIST tool       %secur;                                  >

<!ELEMENT partno     - -       (#PCDATA)                 >

<!ATTLIST partno     %secur;                                  >
```

```
<!ELEMENT modelno   - -       (#PCDATA)          >
<!ATTLIST modelno   %secur;                           >

<!ELEMENT sssn      - -       (#PCDATA)          >
<!ATTLIST sssn      %secur;                           >

<!ELEMENT figindex  - -       (#PCDATA)          >
<!ATTLIST figindex  %secur;                           >

<!ELEMENT testeq    - -       (#PCDATA)          >
<!ATTLIST testeq    %secur;                           >

<!ELEMENT material  - -       (#PCDATA)          >
<!ATTLIST material  %secur;                           >

<!ELEMENT torquevl  - -       (#PCDATA)          >
<!ATTLIST torquevl  %secur;                           >

<!ELEMENT stchg          - o       EMPTY                >
<!ATTLIST stchg          level          NUMBER         #IMPLIED
               status         (add | delete) #IMPLIED
               %secur;                          >

<!ELEMENT endchg    - o       EMPTY                   >
```

```
<!ATTLIST endchg    level          NUMBER         #IMPLIED
               status         (add | delete) #IMPLIED
               %secur;                            >


<!ELEMENT subscrpt  - -       (#PCDATA,
                         subscrpt?,
                         supscrpt?)               >

<!ATTLIST subscrpt  %secur;                            >


<!ELEMENT supscrpt  - -       (#PCDATA,
                         subscrpt?,
                         supscrpt?)               >
<!ATTLIST supscrpt  %secur;                            >


]>
```

APPENDIX E

OVERHAUL INSTRUCTIONS DOCUMENT TYPE DEFINITION

```
<!DOCTYPE overhaul [

<!ENTITY % yesorno "NUMBER">

<!ENTITY % bodyatt  "id        ID        #IMPLIED
               inschlvl  NUTOKEN        #IMPLIED
               delchlvl  NUTOKEN        #IMPLIED
               label          NMTOKEN        #IMPLIED
               texttype  NUMBER         #IMPLIED
               itemid         NMTOKEN        #IMPLIED
               config         NUTOKENS  #IMPLIED
               skilltrk  CDATA          #IMPLIED
               hcp       %yesorno; '0'
               xref      IDREF          #IMPLIED" >


<!ENTITY % colfmt   "colnum         NUMBER         #REQUIRED
               datatype  CDATA          #IMPLIED
               left      %yesorno; '0'
               right          %yesorno; '0'
               center         %yesorno; '0'
               char      CDATA          #IMPLIED
               width          NUMBER         #IMPLIED
               leader         CDATA          #IMPLIED
               colsep         %yesorno; #IMPLIED
               rowhead        %yesorno; '0' "          >

<!ENTITY % contdef  "TBD" >

<!ENTITY % def           "" >


<!ENTITY % list          "seqlist | randlist | deflist"
     >


<!ENTITY % secur    "security (uc | c |
                          s | ts)  #IMPLIED
               restrict  NMTOKENS  #IMPLIED
               release        NMTOKENS  #IMPLIED
               codeword  NMTOKENS  #IMPLIED
               scilevel  %yesorno  '0'
               diglyph        NMTOKENS  #IMPLIED" >

<!ENTITY % spcpara  "(warning*, caution*, note*)"          >

<!ENTITY % change   "stchg | endchg"                       >
<!ENTITY % asyntxt  "stemg | endemg | %change; |
               stemph | endemph | config"              >
<!ENTITY % nums          "partno | modelno | ssan | figindex"
     >
<!ENTITY % text     "%asyntxt; |
               %nums; | tool | testeq |
               material | torquevl |
               xref | graphic |
               subscrpt | supscrpt"                    >

<                    !                    -                    -
****************************************************************
     -->
<!--           End Entities                           -->
<                    !                    -                    -
****************************************************************
     -->


<                    !                    -                    -
****************************************************************
     -->
<!-- Definitions of Elements and their Attributes           -->
<                    !                    -                    -
****************************************************************
     -->

<!ELEMENT overhaul  - -      (front, body)                 >
<!--ELENOTE    overhaul  Technical Manual : Depot Level
                    Overhaul Instructions          -->

<!ATTLIST overhaul  status         (revision |
                         change |
                         prelim |
                         draft |
                         formal)       "formal"
                    revno          NUMBER         "0"
                    chgno          NUMBER         "0"
                      %secur;                               >
<!--ATTNOTE overhaul     status         "This is the status of the

                         document at the time of
                         composition, and is usually
                         reflected on the title page."
               revno          "Revision number of the doc."
               chgno          "Change level of the document."
               %secur;        "defined in entities          -->
```

```
<!ELEMENT front              - -        (idinfo,
                             lep,
                             verstat?,
                             contents,
                             loi, lot,
                             intro?)                  >

<!ATTLIST front              %secur;                            >


<!ELEMENT idinfo     - -          (pubno+,
                             prepubno?,
                             %spcpara;,
                             titleblk,
                             (mfr, contrno)+,
                             docmfr?,
                             notice+,
                             pubdate,
                             chgdate?)            >

<!ATTLIST idinfo     %secur;                           >


<!ELEMENT pubno              - -        (user, docno)           >

<!ATTLIST pubno              %secur;                            >


<!ELEMENT prepubno   - -          (user, docno)            >

<!ATTLIST prepubno   %secur;                           >


<!ELEMENT user       - -          (#PCDATA)           >

<!ATTLIST user       %secur;                           >


<!ELEMENT titleblk   - -          (doctype,
                             maintlvl*,
                             prtitle)             >
```

```
<!ATTLIST titleblk  %secur;                                    >


<!ELEMENT doctype    - -        (#PCDATA) +(%text;) >
<!ATTLIST doctype   %secur;                                    >


<!ELEMENT maintlvl   - -        (#PCDATA) +(%text;)                 >
<!ATTLIST maintlvl  %secur;                                    >

<!ELEMENT prtitle    - -        (nomen, type?)                 >
<!ATTLIST prtitle   %secur;                                    >

<!ELEMENT nomen            - -        (#PCDATA)                >
<!ATTLIST nomen           %secur;                                   >

<!ELEMENT pteqno     - -        (#PCDATA)            >
<!ATTLIST pteqno    %secur;                                    >

<!ELEMENT applic     - -        (#PCDATA)            >
<!ATTLIST applic    %secur;                                    >

<!ELEMENT component - -         (#PCDATA)            >
<!ATTLIST component %secur;                                    >

<!ELEMENT quantity   - -        (#PCDATA)            >
<!ATTLIST quantity  %secur;                                    >

<!ELEMENT refno            - -        (#PCDATA)                >
<!ATTLIST refno           %secur;                                   >
```

```
<!ELEMENT figno          - -       (#PCDATA)           >
<!ATTLIST figno          %secur;                            >

<!ELEMENT descrip   - -       (#PCDATA) +(%text;) >
<!ATTLIST descrip   %secur;                            >

<!ELEMENT minimum   - -       (#PCDATA) +(%text;) >
<!ATTLIST minimum   %secur;                            >

<!ELEMENT maximum   - -       (#PCDATA) +(%text;) >
<!ATTLIST maximum   %secur;                            >

<!ELEMENT replace   - -       (#PCDATA) +(%text;) >
<!ATTLIST replace   %secur;                            >

<!ELEMENT type      - -       (#PCDATA)           >
<!ATTLIST type      %secur;                            >

<!ELEMENT charac    - -       (para+)                  >
<!ATTLIST charac    %secur;                            >

<!ELEMENT general   - -       (para+)                  >
<!ATTLIST general   %secur;                            >

<!ELEMENT purpose   - -       (para+)                  >
<!ATTLIST purpose   %secur;                            >

<!ELEMENT factdata  - -       (para+)                  >
```

```
<!ATTLIST factdata  %secur;                            >

<!ELEMENT mfr       - -       (#PCDATA)           >

<!ATTLIST mfr       %secur;                            >

<!ELEMENT docmfr    - -       (#PCDATA)           >

<!ATTLIST docmfr    %secur;                            >


<!ELEMENT contrno   - -       (#PCDATA)           >

<!ATTLIST contrno   %secur;                            >


<!ELEMENT notice    - -       ((#PCDATA) |
                         (%list;))*     +(%text;) >

<!ATTLIST notice    type      (dist     |
                         auth |
                         offic     |
                         branch    |
                         pgclass   |
                         discl     |
                         supersed|
                         effdate   |
                         suppl     |
                         nopg |
                         noclaspg) #IMPLIED
               %secur;                            >


<!ELEMENT pubdate   - -       (#PCDATA)           >

<!ATTLIST pubdate   %secur;                            >


<!ELEMENT chgdate   - -       (#PCDATA)           >

<!ATTLIST chgdate   %secur;                            >
```

```
<!ELEMENT date      - -      (#PCDATA)            >

<!ATTLIST date      %secur;                              >


<!ELEMENT lep       - o      EMPTY                       >

<!ELEMENT loi       - o      EMPTY                       >

<!ELEMENT lot       - o      EMPTY                       >

<!ELEMENT verstat   - -      (para,
                         (para |
                         %list; |
                         (%spcpara;))*)           >

<!ATTLIST verstat   %secur;                              >


<!ELEMENT contents  - o      EMPTY                       >


<!ELEMENT intro          - -      (title,
                         (title?,
                         (para |
                         abbrsect |
                         symsect |
                         tctorec |
                         (%list;)  )+ )+ )        >

<!ATTLIST intro          %secur;                              >


<!ELEMENT seqlist   - -      (title?, item*)              >

<!ATTLIST seqlist   label         CDATA           #IMPLIED
               %secur;                            >
```

```
<!ELEMENT randlist  - -       (title?, item*)            >

<!ATTLIST randlist  %secur;                         >


<!ELEMENT deflist   - -       (term, def)+              >

<!ATTLIST deflist   %secur;                         >


<!ELEMENT item       - -      (#PCDATA) +(%text;)        >
<!ATTLIST item      %bodyatt;
               %secur;                         >


<!ELEMENT symsect   - -       (deflist)            >

<!ATTLIST symsect   %secur;                         >


<!ELEMENT abbrsect  - -       (deflist)            >

<!ATTLIST abbrsect  %secur;                         >


<!ELEMENT term      - -       (#PCDATA) +(%text;)        >

<!ATTLIST term      %secur;                         >


<!ELEMENT tctorec   - -       (pubno,
                         title,
                         date?)+              >

<!ATTLIST tctorec   %secur;                         >


<!ELEMENT title          - -       (#PCDATA) +(%text;)        >

<!ATTLIST title          %secur;
```

```
               texttype  NUMBER         #IMPLIED  >

<!ELEMENT def       - -       ((#PCDATA) |
                         table)+        -(%list;)
                                   +(%text;) >

<!ATTLIST def       %secur;                                >


<!ELEMENT para      - -       (title?,
                         (#PCDATA),
                         (%list; |
                         (%spcpara;))*)+          +(%text;)      >

<!ATTLIST para      %bodyatt;
               %secur;                            >


<!ELEMENT chart          - -       ( title,
                         ( tabdef |
                         stdtable ),
                         thead?,
                         tbody )             >

<!ATTLIST chart          %bodyatt;
               %secur;                            >


<!ELEMENT table          - -       ( title,
                         (tabdef |
                         stdtable ),
                         thead?,
                         tbody)          +(warning |
                                   caution |
                                   note)          >

<!ATTLIST table          %bodyatt;
               %secur;                            >


<!ELEMENT stdtable   - o      EMPTY                    >
```

```
<!ATTLIST stdtable  name        ENTITY            #REQUIRED
               %secur;                                >


<!ELEMENT tabdef     - -        (colbddef)*                 >

<!ATTLIST tabdef     siderule   %yesorno; "0"
               colsep           %yesorno; "0"
               rowsep           %yesorno; "0"
               orient           (port | land)  "port"
               width            (page | column) "page"
               %secur;                                >


<!ELEMENT colbddef  - o         EMPTY                       >

<!ATTLIST colbddef  %colfmt;                          >


<!ELEMENT thead             - -      (row+)                      >

<!ATTLIST thead             %secur;                              >


<!ELEMENT tbody             - -      ((row |
                     %spcpara;)+)                 >

<!ATTLIST tbody             %secur;                              >


<!ELEMENT row        - -        (entry+)              >

<!ATTLIST row        number          NUMBER         #REQUIRED
               rowsep           %yesorno; #IMPLIED
               %secur;                                >


<!ELEMENT entry             - -      (#PCDATA) -(figure |
                                     table |
                                     chart)
                                     +(%text;) >
```

```
<!ATTLIST entry          col        NUTOKEN         #REQUIRED
               row       NMTOKEN         #CURRENT
               rotate         NUMBER          "0"
               %secur;                              >


<!ELEMENT figure     - -       (graphic+,
                     title,
                     legend?)            >

<!ATTLIST figure    %bodyatt;
               %secur;                              >


<!ELEMENT legend     - -       (deflist)            >

<!ATTLIST legend    %secur;                              >


<!ELEMENT body       - -       (geninfo,
                      toolsect,
                      disassem,
                      clean,
                      irr,
                      assembly,
                      accessor?,
                      testing,
                      tol, ipb?,
                      dds?)                 >
<!-- The body of an Depot maintenance instruction set should
consist
     of a General Information section and a Special tool section

     followed by a Disassembly Section, Cleaning section,
Inspection,
     Repair and Replacement Section, Assembly Section, an optional

     Accessories section, Testing section, and Table of Limits
section.
     These sections will be followed optionally by an Illustrated

     Parts Breakdown and a Difference Data Sheet.
          -->
```

```
<!ATTLIST body      %secur;                                  >


<!ELEMENT geninfo   - -        (general, purpose,
                           charac, factdata)  +(figure |
                                             chart |
                                             table)          >
<!--ELENOTE geninfo      This section gives an overall description

                    of the system                        -->
<!ATTLIST geninfo   %secur;                                  >


<!ELEMENT toolsect  - -        (geninfo,
                           tooleqpt)          +(figure)>

<!--ELENOTE toolsect           This section is for special tools and

                    test equipment                       -->
<!ATTLIST toolsect  %secur;                                  >


<!ELEMENT tooleqpt  - -        (pteqno,
                           figindex,
                           nomen)+            >

<!--ELENOTE tooleqpt           This is a tabular form list with the

                    following format:
               SPECIAL TOOLS and TEST EQUIPMENT
Tool/equipment             Figure &
Part Number                Index No.          Nomenclature

-->
<!ATTLIST tooleqpt  %secur;                                  >

<!ELEMENT disassem  - -        (general,
                           subsect+)     +(figure |
                                             chart |
                                             table)          >

<!ELEMENT clean            - -     (general,
                           subsect+ )         +(figure |
                                             chart |
                                             table)     >
<!ATTLIST clean            %bodyatt;
               %secur;                                  >
```

```
<!ELEMENT irr        - -      (general,
                          subsect+ )        +(figure |
                                         chart |
                                         table)    >
<!ATTLIST irr        %bodyatt;
               %secur;                             >

<!ELEMENT assembly   - -      (general,
                          subsect+ )        +(figure |
                                         chart |
                                         table)    >
<!ATTLIST assembly   %bodyatt;
               %secur;                             >

<!ELEMENT accessor   - -      (disassem,
                          clean,
                          irr,
                          assembly,
                          testing )         +(figure |
                                         chart |
                                         table)    >
<!ATTLIST accessor   %bodyatt;
               %secur;                             >

<!ELEMENT testing    - -      (general,
                          subsect+ )        +(figure |
                                         chart |
                                         table)    >
<!ATTLIST testing    %bodyatt;
               %secur;                             >


<!ELEMENT tol        - -      (general,
                          limittbl,
                          torquetbl)        +(figure)>

<!--ELENOTE tol               This section is for distinction
                     of limit points and lubrication
                     system                   -->
<!ATTLIST tol        %secur;                            >


<!ELEMENT limittbl   - -      (refno,
                          figno,
                          descrip,
                          minimum,
                          maximum,
                          replace)+                >
```

```
<!ATTLIST limittbl  %secur;                                    >


<!ELEMENT torquetbl - -         (applic,
                          component,
                          quantity,
                          torquevl)+                      >
<!ATTLIST torquetbl %secur;                                    >


<!ELEMENT ipb       - o         EMPTY                          >
<!-- put in the IPB definition we came up with before
-->
<!--ELENOTE ipb          This section consists of a Illustrated
               Parts Breakdown in accordance with
               MIL-M-38807.                      -->


<!ELEMENT dds       - -         (general,
                          subsect+,
                          limittbl?,
                          torquetbl?)     +(figure |
                                        chart |
                                        table)    >
<!ATTLIST dds       %bodyatt;
               %secur;                                    >


<!ELEMENT subsect   - -         (title,
                         para+,
                         taskinfo*,
                         para* )                          >
<!ATTLIST subsect   %bodyatt;
               %secur;                                    >



<!ELEMENT taskinfo  - -         (title?,
                         para*,
                         step1+,
                         para* )                          >
<!ATTLIST taskinfo  %bodyatt;
               %secur;                                    >


<!ELEMENT step1          - -         ((%spcpara;),
                         (#PCDATA),
                         (step2)*,
                         (%list;)?)     +(figure |
                                    %text;) >
<!ATTLIST step1          %bodyatt;
               %secur;
               ipi       NAME      #IMPLIED
               person         NAME      #IMPLIED  >


<!ELEMENT step2          - -       ((%spcpara;),
                         (#PCDATA),
                         (step3)*,
                         (%list;)?)     +(%text;) >
<!ATTLIST step2          %bodyatt;
               %secur;
               ipi       NAME      #IMPLIED
               person         NAME      #IMPLIED  >


<!ELEMENT step3          - -       ( (%spcpara;),
                         (#PCDATA),
                         (%list;)?)     +(%text;) >

<!ATTLIST step3          %bodyatt;
               %secur;
               ipi       NAME      #IMPLIED
               person         NAME      #IMPLIED  >



<!ELEMENT docno          - -       (#PCDATA)           >

<!ATTLIST docno          %secur;                            >


<!ELEMENT warning   - -       (graphic?,
                         (para*) )           >

<!ATTLIST warning   name      ENTITY         #CONREF
               type      NAME      #IMPLIED
               xref      IDREF          #IMPLIED
               vital          (y | n)        "n"
               %secur;                        >
```

```
<!ELEMENT caution   - -       (graphic?,
                         (para*) )           >
<!ATTLIST caution   name      ENTITY         #CONREF
               type      NAME      #IMPLIED
               xref      IDREF          #IMPLIED
               %secur;                             >


<!ELEMENT note      - -       (graphic?,
                         (para*) )           >
<!ATTLIST note      name      ENTITY         #CONREF
               type      NAME      #IMPLIED
               xref      IDREF          #IMPLIED
               %secur;                             >


<!ELEMENT stemg          - o       EMPTY                    >

<!ELEMENT endemg    - o       EMPTY                    >

<!ELEMENT stemph    - o       EMPTY                    >
<!ATTLIST stemph    type      NUMBER         #REQUIRED >

<!ELEMENT endemph   - o       EMPTY                    >
<!ATTLIST endemph   type      NUMBER         #REQUIRED >

<!ELEMENT config    - o       EMPTY                    >

<!ATTLIST config    code      CDATA          #REQUIRED
               %secur;                             >


<!ELEMENT xref      - o       EMPTY                    >
<!ATTLIST xref      xref      CDATA          #REQUIRED
               %secur;                             >


<!ELEMENT graphic   - o       EMPTY                    >
<!ATTLIST graphic   boardno        CDATA          #REQUIRED
               width          NUTOKEN        #IMPLIED
```

```
                      depth         NUTOKEN        #IMPLIED
                      window        CDATA          #IMPLIED
                      hscale        NUTOKEN        #IMPLIED
                      vscale        NUTOKEN        #IMPLIED
                      text      CDATA          #IMPLIED
                      rotation  NUMBER         "0"
                      %secur;                            >

<!ELEMENT tool        - -       (#PCDATA)          >

<!ATTLIST tool        %secur;                            >

<!ELEMENT partno      - -       (#PCDATA)          >

<!ATTLIST partno      %secur;                            >

<!ELEMENT modelno     - -       (#PCDATA)          >

<!ATTLIST modelno     %secur;                            >

<!ELEMENT sssn        - -       (#PCDATA)          >

<!ATTLIST sssn        %secur;                            >

<!ELEMENT figindex    - -       (#PCDATA)          >

<!ATTLIST figindex    %secur;                            >

<!ELEMENT testeq      - -       (#PCDATA)          >

<!ATTLIST testeq      %secur;                            >

<!ELEMENT material    - -       (#PCDATA)          >

<!ATTLIST material    %secur;                            >
```

```
<!ELEMENT torquevl  - -      (#PCDATA)        >

<!ATTLIST torquevl  %secur;                        >


<!ELEMENT stchg          - o       EMPTY                    >

<!ATTLIST stchg          level          NUMBER         #IMPLIED
               status         (add | delete) #IMPLIED
               %secur;                  >


<!ELEMENT endchg    - o       EMPTY                    >

<!ATTLIST endchg    level          NUMBER         #IMPLIED
               status         (add | delete) #IMPLIED
               %secur;                  >


<!ELEMENT subscrpt  - -      (#PCDATA,
                    subscrpt?,
                    supscrpt?)             >

<!ATTLIST subscrpt  %secur;                        >


<!ELEMENT supscrpt  - -      (#PCDATA,
                    subscrpt?,
                    supscrpt?)             >
<!ATTLIST supscrpt  %secur;                        >


]>
^Z
```